SN74ACT8832
CMOS 32-Bit Registered ALU

- **50-ns** Cycle Time
- **Low-Power EPIC™** CMOS
- **Three-Port I/O** Architecture
- **64-Word by 36-Bit** Register File
- Simultaneous ALU and Register **Operations**
- Configurable as Quad 8-Bit or Dual 16-Bit Single Instruction, Multiple Data Machine
- Parity Generation/Checking

The SN74ACT8832 is a 32-bit registered ALU that can operate at 20 MHz and 20 MIPS (million instructions per second). Most instructions can be performed in a single cycle. The 'ACT8832 was designed for applications that require high-speed logical, arithmetic, and shift operations and bit/byte manipulations.

The 'ACT8832 can act as host CPU or can accelerate a host microprocessor. In high-performance graphics systems, the 'ACT8832 generates display-list memory addresses and controls the display buffer. In I/O controller applications, the 'ACT8832 performs high-speed comparisons to initialize and end data transfers.

A three-operand, 64-word by 36-bit register file allows the 'ACT8832 to create an instruction and store the previous result in a single cycle.

EPIC is a trademark of Texas Instruments Incorporated.

Contents

Contents (Continued)

Contents (Concluded)

3

SN74ACT8832

List of Illustrations

List of Tables

Introduction

The SN74ACT8832 Registered Arithmetic/Logic Unit (ALU) holds a primary position in the Texas Instruments family of innovative 32-bit LSI devices. Compatible with the SN74AS888 architecture and instruction set, the 'ACT8832 performs as a high-speed microprogrammable 32-bit registered ALU which can also be configured to operate as two 16-bit ALUs or four 8-bit ALUs in single-instruction, multiple-data (SIMD) mode.

Besides introducing the 'ACT8832, this section discusses basic concepts of microprogrammed architecture and the support tools available for system development. Details of the 'ACT8832 architecture and instruction set are presented. Pin descriptions and assignments for the 'ACT8832 are also presented.

Understanding Microprogrammed Architecture

Figure 1 shows a simple microprogrammed system. The three basic components are an arithmetic/logic unit, a microsequencer, and a memory. The program that resides in this memory is commonly called the microprogram, while the memory itself is referred to as a micromemory or control store. The ALU performs all the required operations on data brought in from the external environment (main memory or peripherals, for example). The sequencer is dedicated to generating the next micromemory address from which a microinstruction is to be fetched. The sequencer and the ALU operate in parallel so that data processing and next-address generation are carried out concurrently.

The microprogram instruction, or microinstruction, consists of control information to the ALU and the sequencer. The microinstruction consists of a number of fields of code that directly access and control the ALU, registers, bus transceivers, multiplexers, and other system components. This high degree of programmability in a parallel architecture offers greater speed and flexibility than a typical microprocessor, although the microinstruction serves the same purpose as a microprocessor opcode: it specifies control information by which the user is able to implement desired data processing operations in a specified sequence. The microinstruction cycle is synchronized to a system clock by latching the instruction in the microinstruction, or pipeline, register once for each clock cycle. Status results are collected in a status register which the sequencer samples to produce conditional branches within the microprogram.

'ACT8832 Registered ALU

This device comprises a 32-bit ALU, a 64-word by 36-bit register file, two shifters to support double-precision arithmetic, and three independent bidirectional data ports.

The 'ACT8832 is engineered to support high-speed, high-level operations. The ALU's 13 basic arithmetic and logic instructions can be combined with a single- or double-precision shift operation in one instruction cycle. Other instructions support data conversions, bit and byte operations, and other specialized functions.

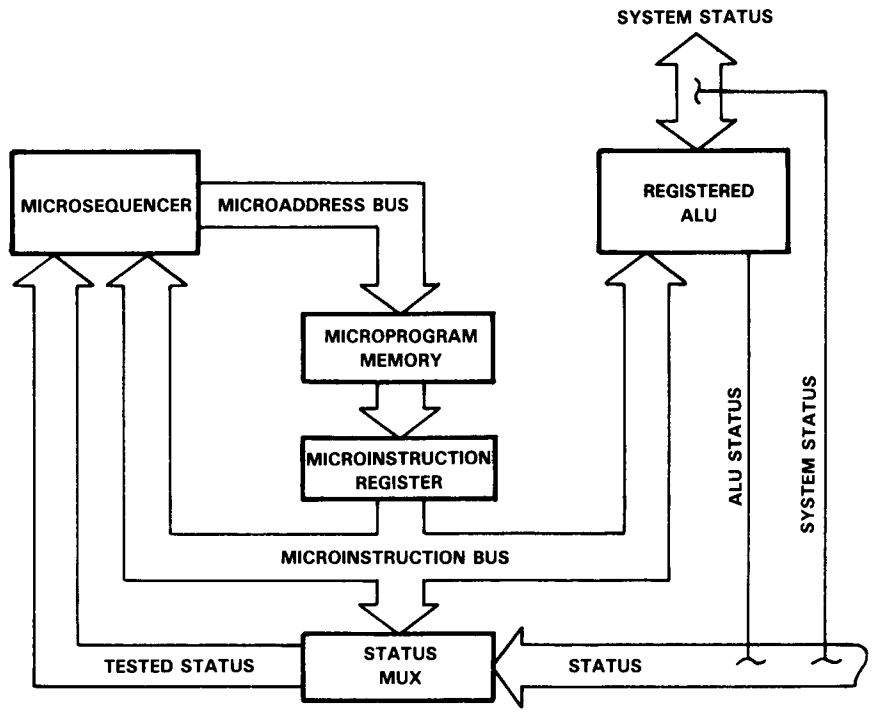


Figure 1. Microprogrammed System Block Diagram

The configuration of this processor enchances processing throughput in arithmetic and radix conversion. Internal generation and testing of status results in fast processing of division and multiplication algorithms. This decision logic is transparent to the user; the reduced overhead assures shorter microprograms, reduced hardware complexity, and shorter software development time.

Support Tools

Texas Instruments has designed a family of low-cost, real-time evaluation modules (EVM) to aid with initial hardware and microcode design. Each EVM is a small self-contained system which provides a convenient means to test and debug simple microcode, allowing software and hardware evaluation of components and their operation.

At present, the 74AS-EVM-8 Bit-Slice Evaluation Module has been completed, and 16- and 32-bit EVMs are in advanced stages of development. EVMs and support tools for other devices in the 'ACT8800 family are also planned for future development.

Design Support

TI's '8832 32-bit registered ALU is supported by a variety of tools developed to aid in design evaluation and verification. These tools will streamline all stages of the design process, from assessing the operation and performance of the '8832 to evaluating a total system application. The tools include a functional model, behavioral model, and microcode development software and hardware. Section 8 of this manual provides specific information on the design tools supporting TI's SN74ACT8800 Family.

Systems Expertise

Texas Instruments VLSI Logic applications group is available to help designers analyze TI's high-performance VLSI products, such as the '8832 32-bit registered ALU. The group works directly with designers to provide ready answers to device-related questions and also prepares a variety of applications documentation.

The group may be reached in Dallas, at (214) 997-3970.

'ACT8832 Pin Descriptions

Pin descriptions and grid allocations for the 'ACT8832 are given on the following pages.

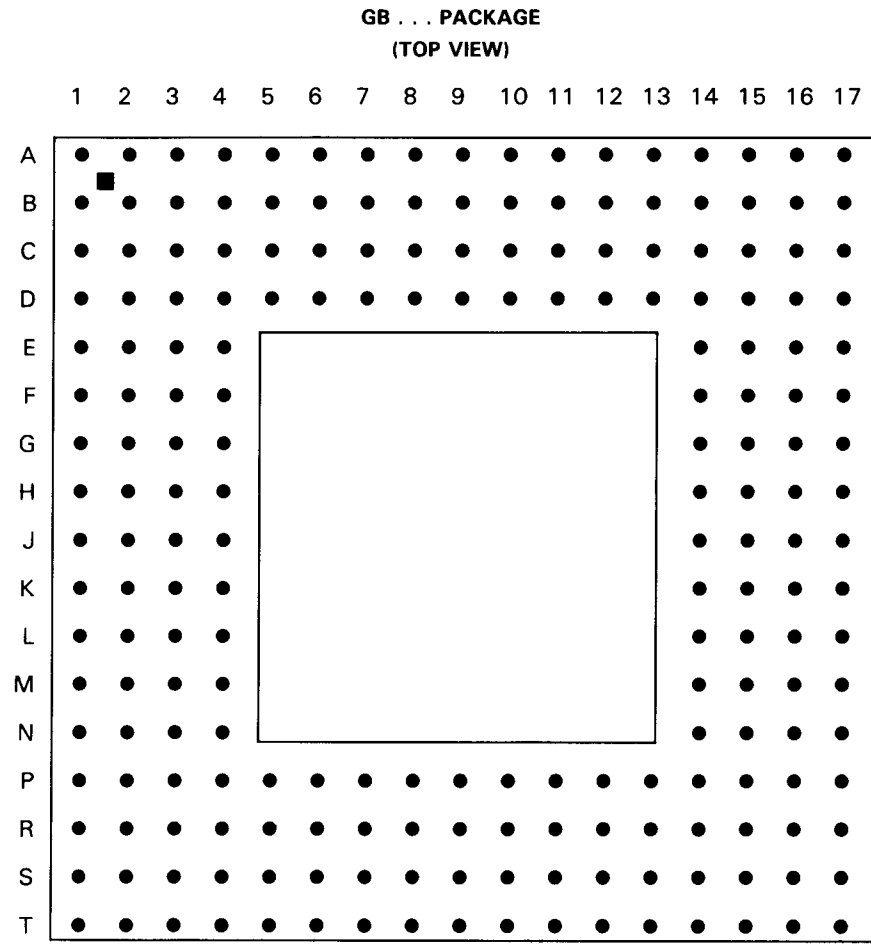


Figure 2. SN74ACT8832 . . . GB Package

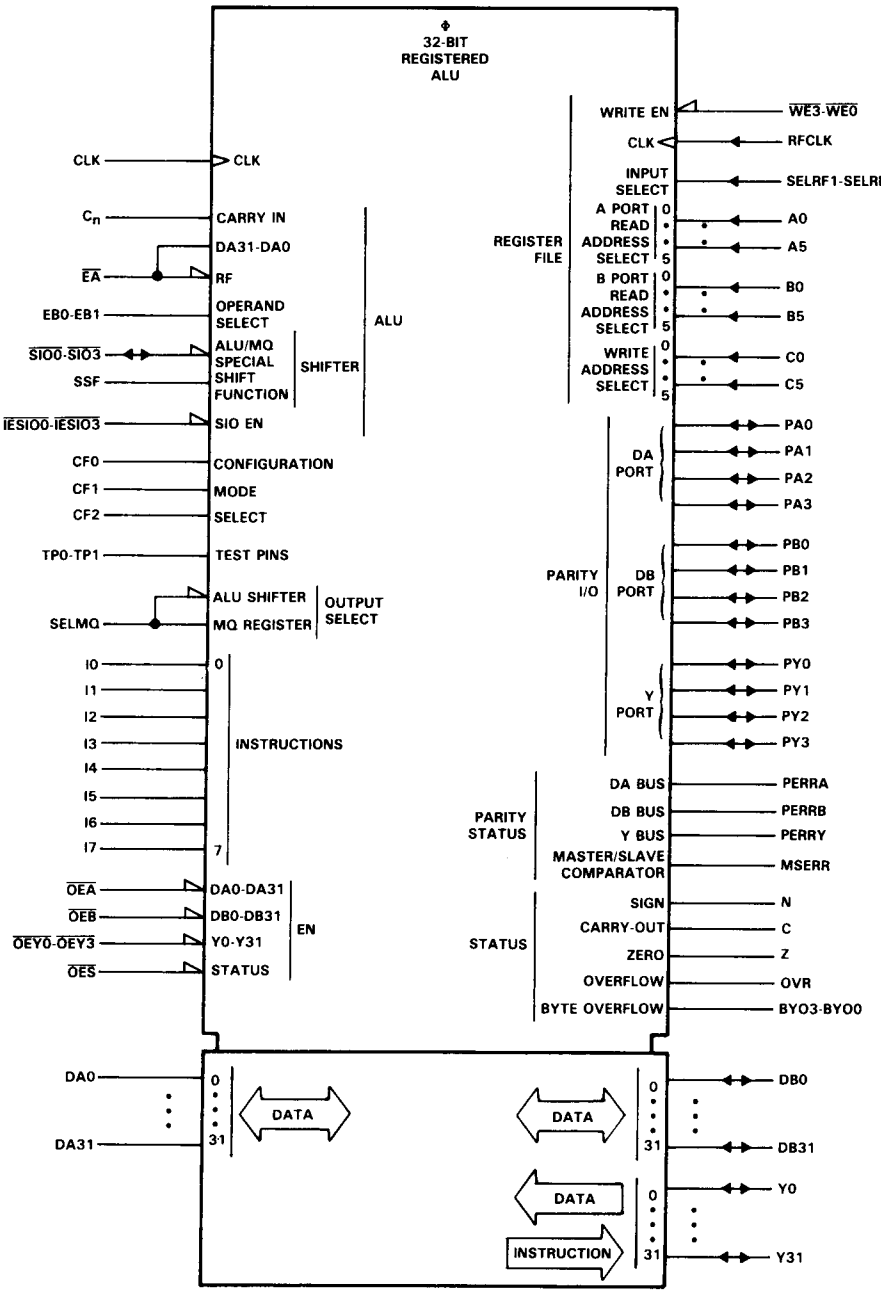


Figure 3. SN74ACT8832 . . . Logic Symbol

Table 1. SN74ACT8832 Pin Grid Allocation

PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
A1	Y7	C2	Y5	E3	Y0	J15	DA28	P1	DA5	S1	DB10
A2	Y13	C3	$\overline{OEY0}$	E4	Y4	J16	DA27	P2	DB8	S2	DB15
A3	Y15	C4	Y9	E14	Y30	J17	DA29	P3	DB12	S3	DA10
A4	BYOF1	C5	Y11	E15	TP0	K1	DB6	P4	DA9	S4	DA13
A5	$\overline{SIO3}$	C6	Y14	E16	I2	K2	DB7	P5	DA15	S5	PERRA
A6	$\overline{SIO2}$	C7	$\overline{OEY1}$	E17	I3	K3	DA0	P6	A5	S6	A3
A7	$\overline{IESIO1}$	C8	GND	F1	EB1	K4	GND	P7	A1	S7	$\overline{WE0}$
A8	$\overline{IESIO0}$	C9	V_{CC}	F2	Cn	K14	GND	P8	V_{CC}	S8	$\overline{WE3}$
A9	$\overline{SIO0}$	C10	C	F3	CLK	K15	DA24	P9	GND	S9	RFCLK
A10	N	C11	PERRY	F4	CF2	K16	DA25	P10	C4	S10	B4
A11	$\overline{OES}$	C12	Y17	F14	$\overline{OEY3}$	K17	DA26	P11	PERRB	S11	B2
A12	SSF	C13	Y22	F15	I1	L1	PB0	P12	GND	S12	C3
A13	Y18	C14	$\overline{OEY2}$	F16	I4	L2	DA2	P13	DB22	S13	C0
A14	Y20	C15	Y28	F17	I6	L3	V_{CC}	P14	DA16	S14	DB17
A15	Y23	C16	PY3	G1	DB0	L4	GND	P15	DA18	S15	DB20
A16	Y24	C17	BYOF3	G2	EA	L14	GND	P16	DA22	S16	DB23
A17	Y25	D1	CF1	G3	EB0	L15	V_{CC}	P17	DB27	S17	DA21
B1	Y6	D2	Y1	G4	GND	L16	DB30	R1	PA0	T1	DB14
B2	BYOF0	D3	Y3	G14	GND	L17	PB3	R2	DB11	T2	DA8
B3	Y10	D4	PY0	G15	I5	M1	DA1	R3	PB1	T3	DA12
B4	Y12	D5	Y8	G16	I7	M2	DA4	R4	DA11	T4	DA14
B5	PY1	D6	GND	G17	PA3	M3	DA7	R5	PA1	T5	$\overline{OEA}$
B6	$\overline{IESIO3}$	D7	GND	H1	DB2	M4	GND	R6	A4	T6	A2
B7	$\overline{IESIO2}$	D8	GND	H2	DB1	M14	PA2	R7	A0	T7	$\overline{WE1}$
B8	$\overline{SIO1}$	D9	V_{CC}	H3	V_{CC}	M15	DB26	R8	$\overline{WE2}$	T8	SELRF1
B9	Z	D10	GND	H4	GND	M16	DB28	R9	V_{CC}	T9	SELRF0
B10	OVR	D11	GND	H14	GND	M17	DB31	R10	B1	T10	B5
B11	MSERR	D12	GND	H15	V_{CC}	N1	DA3	R11	C2	T11	B3
B12	Y16	D13	BYOF2	H16	DA31	N2	DA6	R12	$\overline{OEB}$	T12	B0
B13	Y19	D14	Y27	H17	DA30	N3	DB9	R13	DB18	T13	C5
B14	Y21	D15	Y31	J1	DB3	N4	DB13	R14	DB21	T14	C1
B15	PY2	D16	TP1	J2	DB4	N14	DA19	R15	PB2	T15	DB16
B16	Y26	D17	I0	J3	DB5	N15	DA23	R16	DA20	T16	DB19
B17	Y29	E1	SELMQ	J4	V_{CC}	N16	DB25	R17	DB24	T17	DA17
C1	Y2	E2	CF0	J14	V_{CC}	N17	DB29				

Table 2. SN74ACT8832 Pin Description

PIN NAME	PIN NO.	I/O	DESCRIPTION
A0	R7	I	Register file A port read address select
A1	P7		
A2	T6		
A3	S6		
A4	R6		
A5	P6		
B0	T12	I	Register file B port read address select
B1	R10		
B2	S11		
B3	T11		
B4	S10		
B5	T10		
BYOF0	B2	O	Status signals indicate overflow conditions in certain data bytes
BYOF1	A4		
BYOF2	D13		
BYOF3	C17		
C	C10	O	Status signal representing carry out condition
C0	S13	I	Register file write address select
C1	T14		
C2	R11		
C3	S12		
C4	P10		
C5	T13		
CF0	E2	I	Configuration mode select, single 32-bit, two 16-bit, or four 8-bit ALU's
CF1	D1		
CF2	F4		
Cn	F2	I	ALU carry input
CLK	F3	I	Clocks synchronous registers on positive edge
DA0	K3	I/O	A port data bus. Outputs register data ($\overline{OEA} = 0$) or inputs external data ($\overline{OEA} = 1$).
DA1	M1		
DA2	L2		
DA3	N1		
DA4	M2		
DA5	P1		
DA6	N2		
DA7	M3		
DA8	T2		
DA9	P4		

Table 2. SN74ACT8832 Pin Description (Continued)

PIN NAME	PIN NO.	I/O	DESCRIPTION
DA10	S3	I/O	A port data bus. Outputs register data ($\overline{OEA} = 0$) or inputs external data ($\overline{OEA} = 1$).
DA11	R4		
DA12	T3		
DA13	S4		
DA14	T4		
DA15	P5		
DA16	P14		
DA17	T17		
DA18	P15		
DA19	N14		
DA20	R16		
DA21	S17		
DA22	P16		
DA23	N15		
DA24	K15		
DA25	K16		
DA26	K17		
DA27	J16		
DA28	J15		
DA29	J17		
DA30	H17		
DA31	H16		
DB0	G1	I/O	B port data bus. Outputs register data ($\overline{OEB} = 0$) or used to input external data ($\overline{OEB} = 1$)
DB1	H2		
DB2	H1		
DB3	J1		
DB4	J2		
DB5	J3		
DB6	K1		
DB7	K2		
DB8	P2		
DB9	N3		
DB10	S1		
DB11	R2		
DB12	P3		
DB13	N4		
DB14	T1		
DB15	S2		

Table 2. SN74ACT8832 Pin Description (Continued)

PIN NAME	PIN NO.	I/O	DESCRIPTION
DB16	T15	I/O	B port data bus. Outputs register data ($\overline{OEB} = 0$) or used to input external data ($\overline{OEB} = 1$)
DB17	S14		
DB18	R13		
DB19	T16		
DB20	S15		
DB21	R14		
DB22	P13		
DB23	S16		
DB24	R17		
DB25	N16		
DB26	M15		
DB27	P17		
DB28	M16		
DB29	N17		
DB30	L16		
DB31	M17		
$\overline{EA}$	G2	I	ALU input operand select. High state selects external DA bus and low state selects register file
EB0	G3	I	ALU input operand select. Selects between register file, external DB port and MQ register
EB1	F1		
GND	C8		Ground pins. All ground pins must be used.
GND	D6		
GND	D7		
GND	D8		
GND	D10		
GND	D11		
GND	D12		
GND	G4		
GND	G14		
GND	H4		
GND	H14		
GND	K4		
GND	K14		
GND	L4		
GND	L14		
GND	M4		
GND	P9		
GND	P12		

Table 2. SN74ACT8832 Pin Description (Continued)

PIN NAME	PIN NO.	I/O	DESCRIPTION
I0	D17	I	Instruction input
I1	F15		
I2	E16		
I3	E17		
I4	F16		
I5	G15		
I6	F17		
I7	G16		
$\overline{\text{IESIO0}}$	A8	I	Shift pin enables, increases system speed and reduces bus conflict, active low
$\overline{\text{IESIO1}}$	A7		
$\overline{\text{IESIO2}}$	B7		
IESIO3	B6		
MSERR	B11	O	Master Slave Error pin, indicates error between data at Y output MUX and external Y port
N	A10	O	Output status signal representing sign condition
$\overline{\text{OEA}}$	T5	I	DA bus enable, active low
$\overline{\text{OEB}}$	R12	I	DB bus enable, active low
$\overline{\text{OES}}$	A11	I	Status enable, active low
$\overline{\text{OEY0}}$	C3	I	Y bus output enable, active low
$\overline{\text{OEY1}}$	C7		
$\overline{\text{OEY2}}$	C14		
$\overline{\text{OEY3}}$	F14		
OVR	B10	O	Output status signal represents overflow condition
PA0	R1	I/O	Parity bits port for DA data
PA1	R5		
PA2	M14		
PA3	G17		
PB0	L1	I/O	Parity bits port for DB data
PB1	R3		
PB2	R15		
PB3	L17		
PERRA	S5	O	DA data parity error, signals error if an even parity check fails for any byte
PERRB	P11	O	DB data parity error, signals error if an even parity check fails for any byte
PERRY	C11	O	Y data parity error, signals error if an even parity check fails for any byte

Table 2. SN74ACT8832 Pin Description (Continued)

PIN NAME	PIN NO.	I/O	DESCRIPTION
PY0	D4	I/O	Y port parity data, input and output
PY1	B5		
PY2	B15		
PY3	C16		
RFCLK	S9	I	Register File Clock, allows multiple writes to be performed in one master clock cycle
SELMQ	E1	I	MQ register select, selects output of ALU shifter or MQ register to be placed on Y bus
SELRF0	T9	I	Register File select. Controls selection of the Register File(RF) inputs by the RF MUX
SELRF1	T8		
$\overline{SIO0}$	A9	I/O	Bidirectional shift pin, active low
$\overline{SIO1}$	B8		
$\overline{SIO2}$	A6		
$\overline{SIO3}$	A5		
SSF	A12	I	Special Shift Function, implements conditional shift algorithms
TP0	E15	I	Test pins, supports system testing
TP1	D16		
V_{CC}	C9		Supply voltrage (5 V)
V_{CC}	D9		
V_{CC}	H3		
V_{CC}	H15		
V_{CC}	J4		
V_{CC}	J14		
V_{CC}	L3		
V_{CC}	L15		
V_{CC}	P8		
V_{CC}	R9		
$\overline{WE0}$	S7	I	Register File WRITE ENABLE. Data is written into RF when write enables are low and a low to high Register File Clock (RFCLK) transition occurs. Active low.
$\overline{WE1}$	T7		
$\overline{WE2}$	R8		
$\overline{WE3}$	S8		

Table 2. SN74ACT8832 Pin Description (Concluded)

PIN NAME	PIN NO.	I/O	DESCRIPTION
Y0	E3		
Y1	D2		
Y2	C1		
Y3	D3		
Y4	E4		
Y5	C2		
Y6	B1		
Y7	A1		
Y8	D5		
Y9	C4		
Y10	B3		
Y11	C5		
Y12	B4		
Y13	A2		
Y14	C6		
Y15	A3	I/O	Y port data bus
Y16	B12		
Y17	C12		
Y18	A13		
Y19	B13		
Y20	A14		
Y21	B14		
Y22	C13		
Y23	A15		
Y24	A16		
Y25	A17		
Y26	B16		
Y27	D14		
Y28	C15		
Y29	B17		
Y30	E14		
Y31	D15		
Z	B9	O	Output status signal represents zero condition

'ACT8832 Specification Tables

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC} . . . . . −0.5 V to 6 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) . . . . . ±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) . . . . . ±50 mA
Continuous output current, I_O (V_O = 0 to V_{CC}) . . . . . ±50 mA
Continuous current through V_{CC} or GND pins . . . . . ±100 mA
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

†Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 3. Recommended Operating Conditions

PARAMETER		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5.0	5.5	V
V_{IH}	High-level input voltage	2		V_{CC}	V
V_{IL}	Low-level input voltage	0		0.8	V
I_{OH}	High-level output current			−8	mA
I_{OL}	Low-level output current			8	mA
V_I	Input voltage	0		V_{CC}	V
V_O	Output voltage	0		V_{CC}	V
dt/dv	Input transition rise or fall rate	0		15	ns/V
T_A	Operating free-air temperature	0		70	°C

Table 4. Electrical Characteristics

PARAMETER	TEST CONDITIONS	V_{CC}	T_A = 25°C			SN74ACT8832		UNIT
			MIN	TYP	MAX	MIN	MAX	
V_{OH}	I_{OH} = −20 μA	4.5 V		4.49		4.3		V
		5.5 V		5.49		5.3		
	I_{OH} = −8 mA	4.5 V				3.76		
		5.5 V				4.76		
V_{OL}	I_{OL} = 20 μA	4.5 V		0.01			0.10	V
		5.5 V		0.01			0.10	
	I_{OL} = 8 mA	4.5 V					0.45	
		5.5 V					0.45	
I_I	V_I = V_{CC} or 0	5.5 V					±1	μA
I_{CCQ}	V_I = V_{CC} or 0, I_O	5.5 V					200	μA
C_i	V_I = V_{CC} or 0	5 V					15	pF
ΔI_{CC}†	One input at 3.4 V, other inputs at 0 or V_{CC}	5.5 V					1	mA

Table 5. Register File Write Setup

	PARAMETER	MIN	MAX	UNIT
t_{su}	C5-C0	4		ns
	DA/B32-DA/B0, PA/B3-PA/B0	7		
	I7-I4	13		
	$\overline{OEY3}$-$\overline{OEY0}$	7		
	Y31-Y0	4		
	$\overline{WE3}$-$\overline{WE0}$	4		
	SELRF(DA,DB,PA,PB)	5		
	SELRF(Y)	9		
	SIO	10		
	SELMQ	9		
	$\overline{IESIO3}$-$\overline{IESIO0}$	10		
t_h	ALL	0		

†This is the increase in supply current for each input that is at one of the specified TTL voltge levels rather then 0 V to V_{CC}.

Table 6. Maximum Switching Characteristics

PARAMETER	FROM (INPUT)	TO (OUTPUT)											UNIT
		Y	C	Z	SIO	PERRA/B	N	OVR	PA/B DA/B	PY	PERRY	MSERR	
t_{pd}	A5-A0,B5-B0	36	30	37	28		30	37	16	37			ns
	DA31-DA0,PA3-PA0 DB31-DB0,PB3-PB0	36	25	37	25	20	28	37		37			
	C_n	30	22	31	24		28	28		32			
	$\overline{EA}$	37	28	37	25		31	37		37			
	EB1-EB0	37	28	37	25		31	37		37			
	I7-I0	37	30	37	28		32	37		37			
	CF2-CF0	37	30	37	28		32	37		37			
	$\overline{OEB}$,$\overline{OEA}$								15				
	$\overline{OEY3}$-$\overline{OEY0}$	20								20			
	SELMQ	15								20			
	SIO3-SIO0	15		25			25			27			
	CLK	21								28			
	CLKMQ	37								37			
	RCLK	37	32	37	24		32	37		37			
	$\overline{IESIO3}$-$\overline{IESIO0}$	15		25			25			27			
	SSF	25		30	22		30	22		30			
	Y										15	15	

'ACT8832 Registered ALU

The SN74ACT8832 is a 32-bit registered ALU that can be configured to operate as four 8-bit ALUs, two 16-bit ALUs, or a single 32-bit ALU. The processor instruction set is 100 percent upwardly compatible with the 'AS888 and includes 13 arithmetic and logical functions with 8 conditional shifts, multiplication, division, normalization, add and subtract immediate, bit and byte operations, and data conversions such as BCD, excess-3, and sign magnitude. New instructions permit internal flip-flops controlling BCD and divide operations to be loaded or read.

Additional functions added to the 'ACT8832 include byte parity and master/slave operation. Parity is checked at the three data input ports and generated at the Y output port. The 64-word register file is 36 bits wide to permit storage of the parity bits. Master/slave comparator circuitry is provided at the Y port.

The DA and DB ports can simultaneously input data to the ALU and the 64-word by 36-bit register file. Data and parity from the register file can be output on the DA and DB ports. Results of ALU and shift operations are output at the bidirectional Y port. The Y port can also be used in an input mode to furnish external data to the register file or during master/slave operation as an input to the master/slave comparator.

Three 6-bit address ports allow a two-operand fetch and an operand write to be performed at the register file simultaneously. An MQ shifter and MQ register can also be configured to function independently to implement double-precision 8-bit, 16-bit, and 32-bit shift operations. An internal ALU bypass path increases the speeds of multiply, divide and normalize instructions. The path is also used by 'ACT8832 instructions that permit bits and bytes to be manipulated.

Architecture

Figure 4 is a functional block diagram of the 'ACT8832. Control input signals are summarized in Table 7. Data flow and details of the functional elements are presented in the following paragraphs.

Table 7. 'ACT8832 Response to Control Inputs

SIGNAL	HIGH	LOW
CF2-CF0	See Table 11	See Table 11
$\overline{EA}$	Selects external DA bus	Selects register file
EB1-EB0	See Table 9	See Table 9
$\overline{IESIO3}$-$\overline{IESIO0}$	Normal operation	Force corresponding SIO inputs to high impedance
I7-I0	See Table 15	See Table 15
MQSEL	Selects MQ register	Selects ALU
$\overline{OEA}$	Inhibits DA and PA output	Enables DA and PA output
$\overline{OEB}$	Inhibits DB and PB output	Enables DB and PB output
$\overline{OEY3}$-$\overline{OEY0}$	Inhibits Y and PY outputs	Enables Y and PY outputs
SELRF1-SELRF0	See Table 8	See Table 8
SSF	Selects shifted ALU output	Selects ALU (unshifted) output
TP1-TP0	See Table 14	See Table 14
$\overline{WE3}$-$\overline{WE0}$	Inhibits register file write	Byte enables for register file write (0 = LSB)

Data Flow

As shown in Figure 5, data enters the 'ACT8832 from three primary sources: the bidirectional Y port, which is used in an input mode to pass data to the register file; and the bidirectional DA and DB ports, used to input data to the register file or the R and S buses serving the ALU. Three associated I/O ports (PY, PA, and PB) are provided for associated parity data input and output.

Data is input to the ALU through two multiplexers: R MUX, which selects the R bus operand from the DA port or the register file addressed by A5-A0; and S MUX, which selects data from the DB port, the register file addressed by B5-B0, or the multiplier-quotient (MQ) register.

The result of the ALU operation is passed to the ALU shifter, where it is shifted or passed without shift to the Y bus for possible output from the 'ACT8832 and to the feedback MUX for possible storage in the internal register file. The MQ shifter, which operates in parallel with the ALU shifter, can be loaded from the ALU or the MQ register. The MQ shift result is passed to the MQ register, where it can be routed through the S MUX to the ALU or to the Y MUX for output from the chip.

An internal bypass path allows data from the S MUX to be loaded directly into the ALU shifter or the divide/BCD flip-flops. Data from the divide/BCD flip-flops can be output via the MQ register.

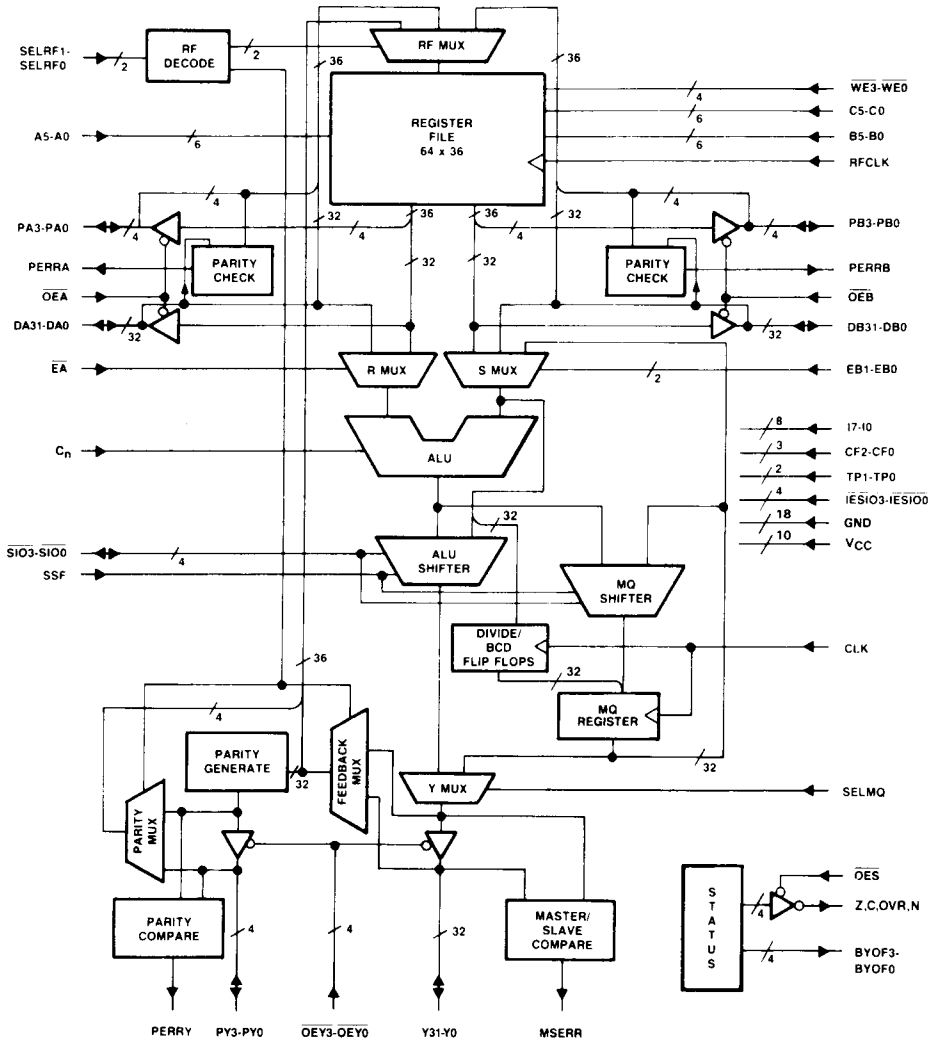


Figure 4. 'ACT8832 32-Bit Registered ALU

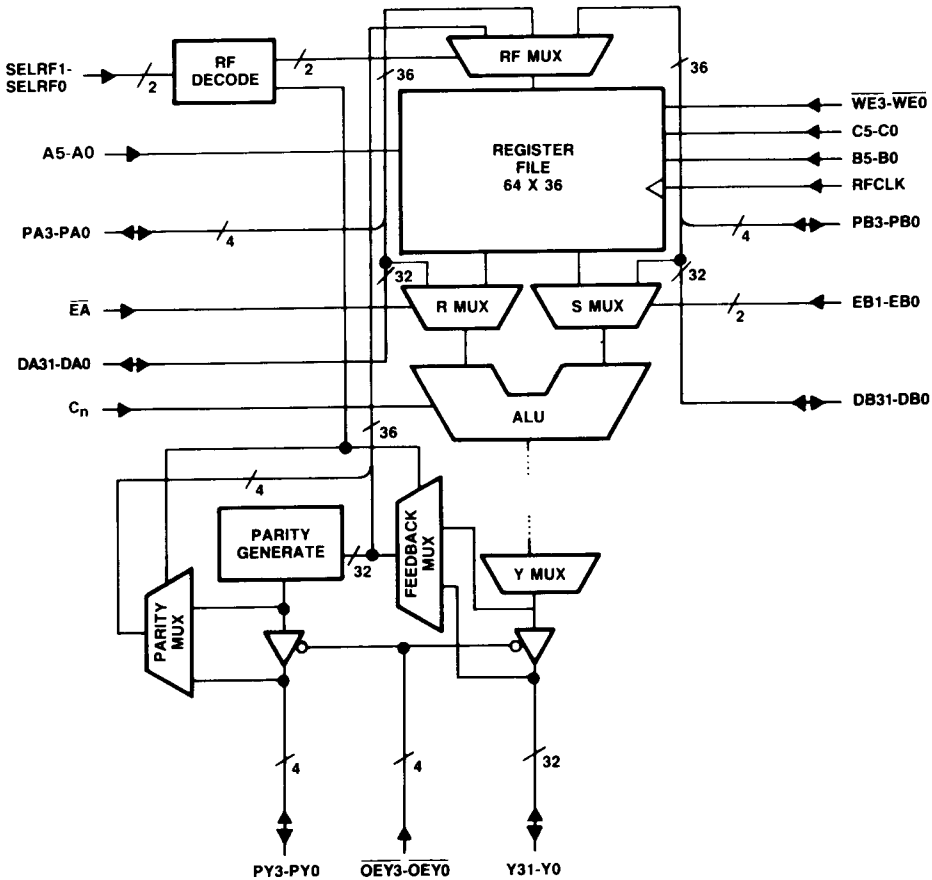


Figure 5. Data I/O

Data can be output from the three bidirectional ports, Y, DA, and DB, and their associated parity ports, PY, PA, and PB. DA and DB can also be used to read ALU input data on the R and S buses for debug or other special purposes.

Architectural Elements

Three-Port Register File

The register file is 36 bits wide, permitting storage of a 32-bit data word with its associated parity bits. The 64 registers are accessed by three address ports. C5-C0 address the destination register during write operations; A5-A0 and B5-B0 address any two registers during read operations. The address buses are also used to furnish

immediate data to the ALU: A3-A0 to provide constant data for the add and subtract immediate instructions; C3-C0 and A3-A0 to provide masks for set, reset, and test bit operations.

Data is written into the register file when the write enable is low and a low-to-high register file clock (RFCLK) transition occurs. The separate register file clock allows multiple writes to be performed in one master clock cycle, allowing processors in multiprocessor environments to update one another's internal register files during a single cycle.

Four write enable inputs are provided to allow separate control of data inputs in a byte-oriented system. $\overline{WE3}$ is the write enable for the most significant byte.

Register file inputs are selected by the RF MUX under the control of two register file select signals, SELRF1 and SELRF0, shown in Table 8 (see also Table 10).

Table 8. RF MUX Select Inputs

SELRF1	SELRF0	SOURCE
0	0	External DA input
0	1	External DB input
1	0	Y-output MUX
1	1	External Y port

R and S Multiplexers

ALU inputs are selected by the R and S multiplexers. Controls which affect operand selection for instructions other than those using constants or masks are shown in Table 9.

Table 9. ALU Source Operand Selects

R-BUS OPERAND SELECT $\overline{EA}$	S-BUS OPERAND SELECT EB1-EB0	RESULT DESTINATION	←SOURCE OPERAND
0		R bus	←Register file addressed by A5-A0
1		R bus	←DA port
	0 0	S bus	←Register file addressed by B5-B0
	1 0	S bus	←DB port
	X 1	S bus	←MQ register

Table 10. Destination Operand Select/Enables

REGISTER FILE WRITE ENABLE $\overline{WE}$	Y BUS OUTPUT ENABLE $\overline{OEY}$	Y MUS SELECT MQSEL	REGISTER FILE SELECT RFSEL1-RFSEL0		DA PORT OUTPUT ENABLE $\overline{OEA}$	DB PORT OUTPUT ENABLE $\overline{OEB}$	RESULT DESTINATION	← SOURCE
1	0	0	X	X			Y/PY	← ALU shifter/parity generate
1	0	1	X	X			Y/PY	← MQ register/parity generate
0	0	0	1	0			Y/PY, RF	← ALU shifter/parity generate
0	0	1	1	0			Y/PY, RF	← MQ register/parity generate
0	1	X	1	1			RF	← External Y/PY
0	X	X	0	0	1	X	RF	← External DA/PA
0	X	X	0	1	X	1	RF	← External DB/PB
					0		DA/PA	← R bus register file output
					1		DA/PA	Hi-Z
						0	DB/PB	← S bus register file output
						1	DB/PB	Hi-Z

Data Input and Output Ports

The DA and DB ports can be used to load the S and/or R multiplexers from an external source or to read S or R bus outputs from the register file. The Y port can be used to load the register file and to output the next address selected by the Y output multiplexer. Tables 9 and 10 describe the MUX and output controls which affect DA, DB, and Y.

ALU

The ALU can perform seven arithmetic and six logical instructions on the two 32-bit operands selected by the R and S multiplexers. It also supports multiplication, division, normalization, bit and byte operations and data conversion, including excess-3 BCD arithmetic. The 'ACT8832 instruction set is summarized in Table 15.

The 'ACT8832 can be configured to operate as a single 32-bit ALU, two 16-bit ALUs, or four 8-bit ALUs (see Figures 6 and 7). It can also be configured to operate on a 32-bit word formed by adding leading zeros to the 12 least significant bits of R bus data. This is useful in certain IBM relative addressing schemes.

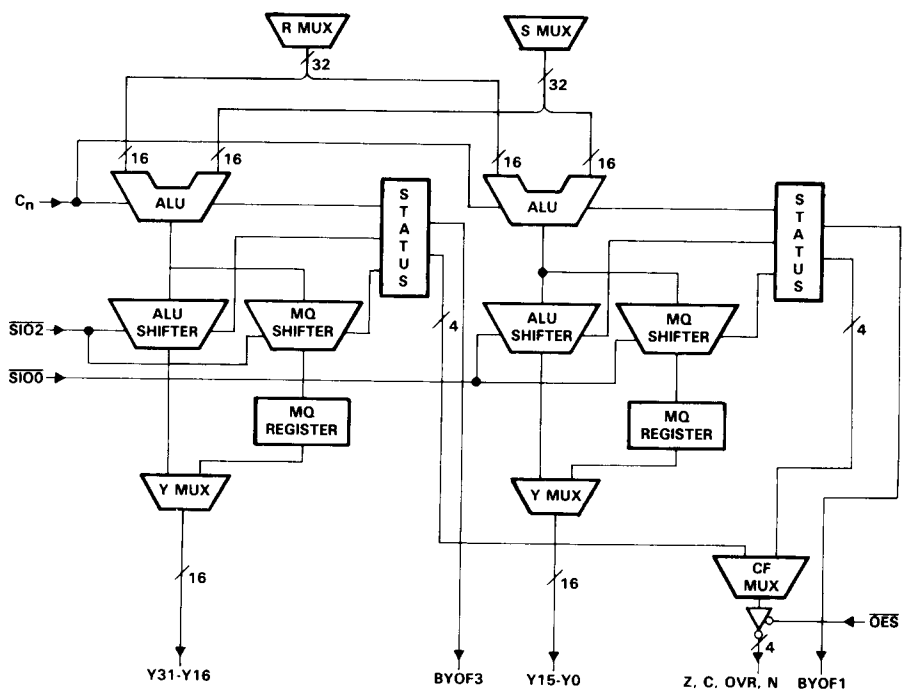


Figure 6. 16-Bit Configuration

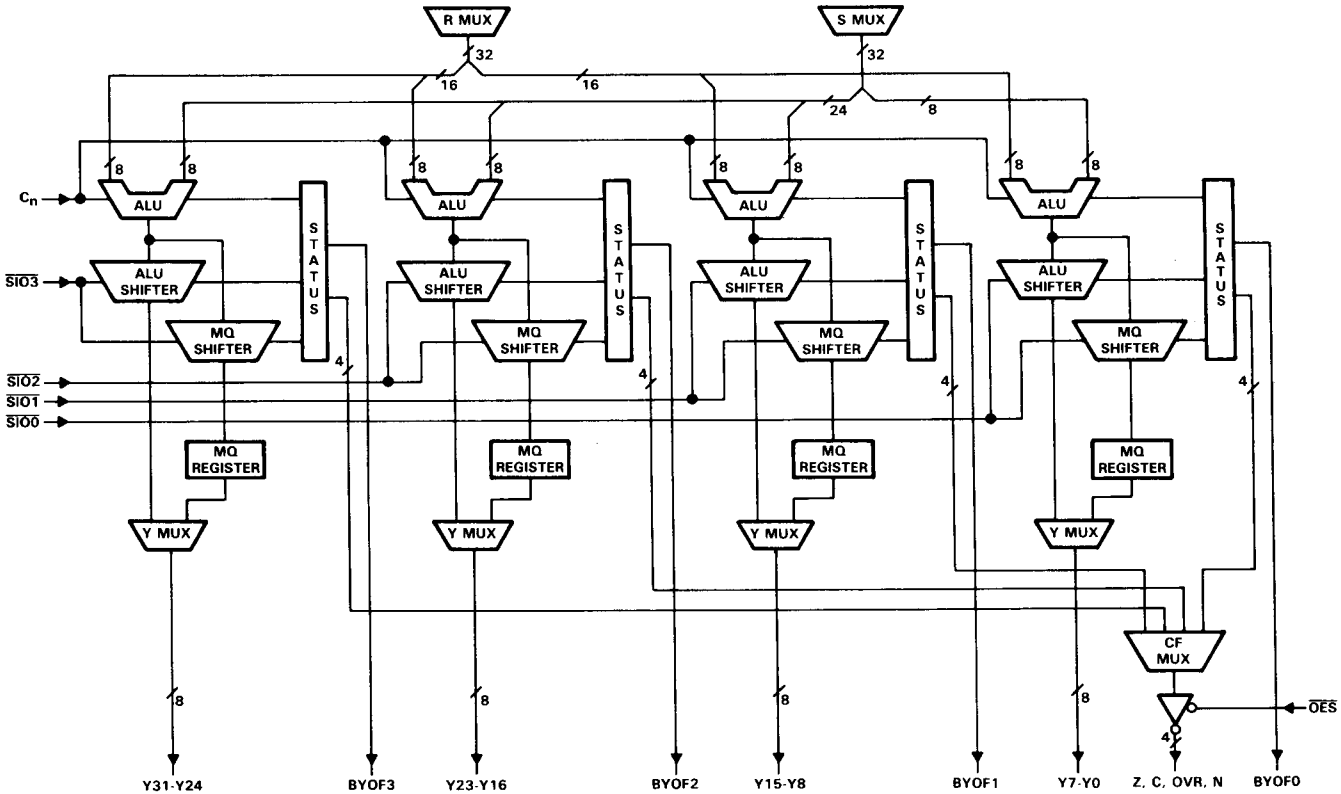


Figure 7. 8-Bit Configuration

Configuration modes are controlled by three CF inputs as shown in Table 11. These signals also select the data from which status signals other than byte overflow will be generated.

Table 11. Configuration Mode Selects

CONTROL INPUTS			MODE SELECTED	DATA FROM WHICH STATUS OTHER THAN BYOF WILL BE GENERATED
CF2	CF1	CF0		
0	0	0	Four 8-bit	Byte 0
0	0	1	Four 8-bit	Byte 1
0	1	0	Four 8-bit	Byte 2
0	1	1	Four 8-bit	Byte 3
1	0	0	Two 16-bit	Least significant 16-bit word
1	0	1	Two 16-bit	Most significant 16-bit word
1	1	0	One 32-bit	32-bit word
1	1	1	Masked 32-bit	32-bit word

ALU and MQ Shifters

The ALU and MQ shifters are used in all of the shift, multiply, divide and normalize functions. They can be used independently for single precision or concurrently for double precision shifts. Shifts can be made conditional, using the Special Shift Function (SSF) pin.

Bidirectional Serial I/O Pins

Four bidirectional $\overline{SIO}$ pins are provided to supply an end fill bit for certain shift instructions. These pins may also be used to read bits that are shifted out of the ALU or MQ shifters during certain instructions. Use of the $\overline{SIO}$ pins as inputs or outputs is summarized in Table 17.

The four pins allow separate control of end fill inputs in configurations other than 32-bit mode (see Table 12 and Figure 4).

Table 12. Data Determining $\overline{SIO}$ Input

SIGNAL	CORRESPONDING WORD, PARTIAL WORD OR BYTE		
	32-BIT MODE	16-BIT MODE	8-BIT MODE
$\overline{SIO3}$	–	–	Byte 3
$\overline{SIO2}$	–	most significant word	Byte 2
$\overline{SIO1}$	–	–	Byte 1
$\overline{SIO0}$	32-bit word	least significant word	Byte 0

To increase system speed and reduce bus conflict, four $\overline{SIO}$ input enables ($\overline{IESIO3}$-$\overline{IESIO0}$) are provided. A low on these enables will override internal pull-up resistor logic and force the corresponding $\overline{SIO}$ pins to the high impedance state required before an input signal can appear on the signal line. If the $\overline{SIO}$ enables are not used, this condition is generated internally in the chip. Use of the enables allow internal decoding to be bypassed, resulting in faster speeds.

The $\overline{IESIO}$s are defaulted to a high because of internal pull-up resistors. When an $\overline{SIO}$ pin is used as an output, a low on its corresponding $\overline{IESIO}$ pin would force $\overline{SIO}$ to a high impedance state. The output would then be lost, but the internal operation of the chip would not be affected.

MQ Register

Data from the MQ shifter is written into the MQ register when a low-to-high transition occurs on clock CLK. The register has specific functions in double precision shifts, multiplication, division and data conversion algorithms and can also be used as a temporary storage register. Data from the register file and the DA and DB buses can be passed to the MQ register through the ALU.

The Y bus contains the output of the ALU shifter if SELMQ is low and the output of the MQ register if SELMQ is high. If $\overline{OEY}$ is low, ALU or MQ shifter output will be passed to the Y port; if $\overline{OEY}$ is high, the Y port becomes an input to the feedback MUX.

Conditional Shift Pin

Conditional shifting algorithms may be implemented using the SSF pin under hardware or firmware control. If the SSF pin is high or floating, the shifted ALU output will be sent to the output buffers. If the SSF pin is pulled low externally, the ALU result will be passed directly to the output buffers, and MQ shifts will be inhibited. Conditional shifting is useful for scaling inputs in data arrays or in signal processing algorithms.

Master/Slave Comparator

A master/slave comparator is provided to compare data bytes from the Y output MUX with data bytes on the external Y port when $\overline{OEY}$ is high. If the data are not equal, a high signal is generated on the master slave error output pin (MSERR). A similar comparator is provided for the Y parity bits.

Divide/BCD Flip-Flops

Internal multiply/divide flip-flops are used by certain multiply and divide instructions to maintain status between instructions. Internal excess-3 BCD flip-flops preserve the carry from each nibble in excess-3 BCD operations. The BCD flip-flops are affected by all instructions except NOP and are cleared when a CLR instruction is executed. The flip-flops can be loaded and read externally using instructions LOADFF and DUMPFF

(see Table 15). This feature permits an iterative arithmetic operation such as multiplication or division to be interrupted immediately so that an external interrupt can be processed.

Status

Eight status output signals are generated by the 'ACT8832. Four signals (BYOF3-BYOF0) indicate overflow conditions in certain data bytes (see Table 13). The others represent sign (N), zero (ZERO), carry-out (Cout) and overflow (OVR). N, ZERO, Cout, and OVR are generated from data selected by the mode configuration controls (CF2-CF0) as shown in Table 11.

Carry-out is evaluated after each ALU operation. Sign and zero status are evaluated after ALU shift operation. Overflow (OVR) is determined by ORing the overflow result from the ALU with the overflow result from the ALU shifter.

Table 13. Data Determining BYOF Outputs

SIGNAL	CORRESPONDING WORD, PARTIAL WORD OR BYTE		
	32-BIT MODE	16-BIT MODE	8-BIT MODE
BYOF3	32-bit word	most significant word	Byte 3
BYOF2	—	—	Byte 2
BYOF1	—	least significant word	Byte 1
BYOF0	—	—	Byte 0

Input Data Parity Check

An even parity check is performed on each byte of input data at the DA, DB and Y ports. The check is performed by counting the number of ones in each byte and its corresponding parity bit. Parity bits are input on PA for DA data, PB for DB data and PYF or Y data. PA0, PB0 and PY0 are the parity bits for the least significant bytes of DA, DB and Y, respectively. If the result of the parity count is odd for any byte, a high appears at the parity error output pin (PERRA for DA data, PERRB for DB data, PERRY for Y data).

Test Pins

Two pins, TP1-TP0, support system testing. These may be used, for example, to place all outputs in a high-impedance state, isolating the chip from the rest of the system (see Table 14).

Table 14. Test Pin Inputs

TP1	TP0	RESULT
0	0	All outputs and I/Os forced low
0	1	All outputs and I/Os forced high
1	0	All outputs and I/Os placed in a high impedance state
1	1	Normal operation (default state)

Instruction Set Overview

Bits I7-I0 are used as instruction inputs to the 'ACT8832. Table 15 lists all instructions, divided into five groups, with their opcodes and mnemonics.

Table 15. 'ACT8832 Instruction Set

GROUP 1 INSTRUCTIONS		
INSTRUCTION BITS I3-I0 (HEX)	MNEMONIC	FUNCTION
0		Used to access Group 4 instructions
1	ADD	R + S + Cn
2	SUBR	$\overline{R}$ + S + Cn
3	SUBS	R + $\overline{S}$ + Cn
4	INCS	S + Cn
5	INCNS	$\overline{S}$ + Cn
6	INCR	R + Cn
7	INCNR	$\overline{R}$ + Cn
8		Used to access Group 3 instructions
9	XOR	R XOR S
A	AND	R AND S
B	OR	R OR S
C	NAND	R NAND S
D	NOR	R NOR S
E	ANDNR	$\overline{R}$ AND S
F		Used to access Group 5 instructions

Table 15. 'ACT8832 Instruction Set (Continued)

GROUP 2 INSTRUCTIONS		
INSTRUCTION BITS I7-I4 (HEX)	**MNEMONIC**	**FUNCTION**
0	SRA	Arithmetic right single precision shift
1	SRAD	Arithmetic right double precision shift
2	SRL	Logical right single precision shift
3	SRLD	Logical right double precision shift
4	SLA	Arithmetic left single precision shift
5	SLAD	Arithmetic left double precision shift
6	SLC	Circular left single precision shift
7	SLCD	Circular left double precision shift
8	SRC	Circular right single precision shift
9	SRCD	Circular right double precision shift
A	MQSRA	Arithmetic right shift MQ register
B	MQSRL	Logical right shift MQ register
C	MQSLL	Logical left shift MQ register
D	MQSLC	Circular left shift MQ register
E	LOADMQ	Load MQ register
F	PASS	Pass ALU to Y

Table 15. 'ACT8832 Instruction Set (Continued)

GROUP 3 INSTRUCTIONS		
INSTRUCTION BITS I7-I0 (HEX)	MNEMONIC	FUNCTION
08	SET1	Set bit 1
18	SET0	Set bit 0
28	TB1	Test bit (one)
38	TB0	Test bit (zero)
48	ABS	Absolute value
58	SMTC	Sign magnitude/two's complement
68	ADDI	Add immediate
78	SUBI	Subtract immediate
88	BADD	Byte add R to S
98	BSUBS	Byte subtract S from R
A8	BSUBR	Byte subtract R from S
B8	BINCS	Byte increment S
C8	BINCNS	Byte increment negative S
D8	BXOR	Byte XOR R and S
E8	BAND	Byte AND R and S
F8	BOR	Byte OR R and S

Table 15. 'ACT8832 Instruction Set (Continued)

GROUP 4 INSTRUCTIONS		
INSTRUCTION BITS I7-I0 (HEX)	MNEMONIC	FUNCTION
00	CRC	Cyclic redundancy character accumulation
10	SEL	Select S or R
20	SNORM	Single length normalize
30	DNORM	Double length normalize
40	DIVRF	Divide remainder fix
50	SDIVQF	Signed divide quotient fix
60	SMULI	Signed multiply iterate
70	SMULT	Signed multiply terminate
80	SDIVIN	Signed divide initialize
90	SDIVIS	Signed divide start
A0	SDIVI	Signed divide iterate
B0	UDIVIS	Unsigned divide start
C0	UDIVI	Unsigned divide iterate
D0	UMULI	Unsigned multiply iterate
E0	SDIVIT	Signed divide terminate
F0	UDIVIT	Unsigned divide terminate

Table 15. 'ACT8832 Instruction Set (Continued)

GROUP 5 INSTRUCTIONS		
INSTRUCTION BITS 17-10 (HEX)	MNEMONIC	FUNCTION
0F	LOADFF	Load divide/BCD flip-flops
1F	CLR	Clear
2F	CLR	Clear
3F	CLR	Clear
4F	CLR	Clear
5F	DUMPFF	Output divide/BCD flip-flops
6F	CLR	Clear
7F	BCDBIN	BCD to binary
8F	EX3BC	Excess-3 byte correction
9F	EX3C	Excess-3 word correction
AF	SDIVO	Signed divide overflow test
BF	CLR	Clear
CF	CLR	Clear
DF	BINEX3	Binary to excess-3
EF	CLR	Clear
FF	NOP	No operation

Group 1, a set of ALU arithmetic and logic operations, can be combined with the user-selected shift operations in Group 2 in one instruction cycle. The other groups contain instructions for bit and byte operations, division and multiplication, data conversion, and other functions such as sorting, normalization and polynomial code accumulation.

Arithmetic/Logic Instructions with Shifts

The seven Group 1 arithmetic instructions operate on data from the R and/or S multiplexers and the carry-in. Carry-out is evaluated after ALU operation; other status pins are evaluated after the accompanying shift operation, when applicable. Group 1 logic instructions do not use carry-in; carry-out is forced to zero.

Possible shift instructions are listed in Group 2. Fourteen single and double precision shifts can be specified, or the ALU result can be passed unshifted to the MQ register or to the specified output destination by using the LOADMQ or PASS instructions. Table 16 lists shift definitions.

When using the shift registers for double precision operations, the least significant half should be placed in the MQ register and the most significant half in the ALU for passage to the ALU shifter. An example of a double-precision shift using the ALU and MQ shifters is given in Figure 8.

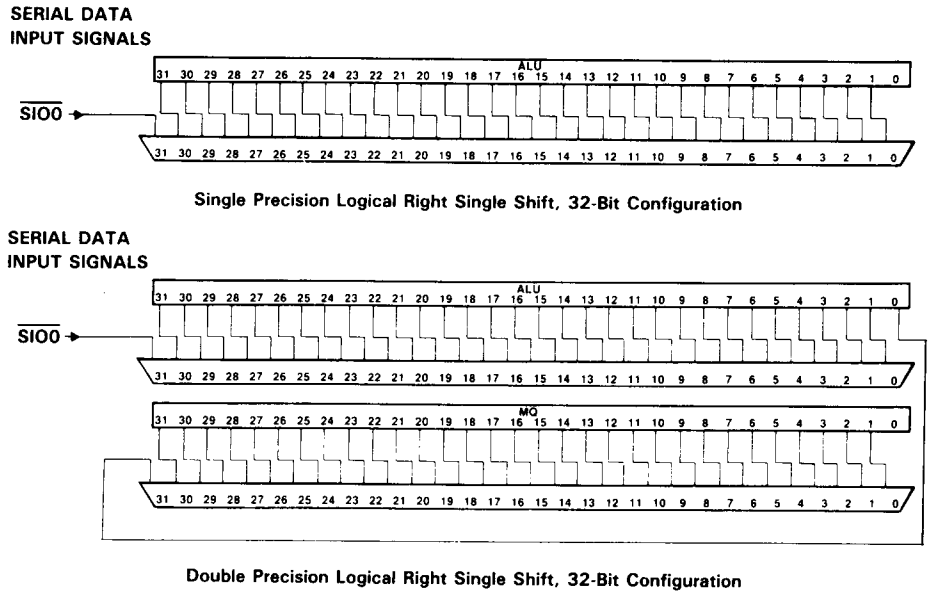


Figure 8. Shift Examples, 32-Bit Configuration

All Group 2 shifts can be made conditional using the conditional shift pin (SSF). If the SSF pin is high or floating, the shifted ALU output will be sent to the output buffers, MQ register, or both. If the SSF pin is pulled low, the ALU result will be passed directly to the output buffers and any MQ shifts will be inhibited.

Table 16. Shift Definitions

SHIFT TYPE	NOTES
Left	Moves a bit one position towards the most significant bit
Right	Moves a bit one position towards the least significant bit
Arithmetic right	Retains the sign unless an overflow occurs, in which case, the sign would be inverted
Arithmetic left	May lose the sign bit if an overflow occurs. Zero is filled into the least significant bit unless the bit is set externally
Circular right	Fills the least significant bit in the most significant bit position
Circular left	Fills the most significant bit in the least significant bit position
Logical right	Fills a zero in the most significant bit position unless the bit is forced to one by placing a zero on an $\overline{SIO}$ pin
Logical left	Fills a zero in the least significant bit position unless the bit is forced to one by placing a zero on an $\overline{SIO}$ pin

The bidirectional $\overline{SIO}$ pins can be used to supply external end fill bits for certain Group 2 shift instructions. When $\overline{SIO}$ is high or floating, a zero is filled, otherwise a 1 is filled Table 17 lists instructions that make use of the $\overline{SIO}$ inputs and identifies input and output functions.

Table 17. Bidirectional SIO Pin Functions

INSTRUCTION BITS I7-I0 (HEX)	$\overline{SIO}$		
	MNEMONIC	I/O	DATA
0*	SRA	O	Shift out
1*	SRAD	O	Shift out
2*	SRL	I	Most significant bit
3*	SRLD	I	Most significant bit
4*	SLA	I	Least significant bit
5*	SLAD	I	Least significant bit
6*	SLC	O	Shifted input to MQ shifter
7*	SLCD	O	Shifted input to MQ shifter
8*	SRC	O	Shifted input to ALU shifter
9*	SRCD	O	Shifted input to ALU shifter
A*	MQSRA	O	Shift out
B*	MQSRL	I	Most significant bit
C*	MQSLL	I	Least significant bit
D*	MQSLC	O	Shifted input to MQ shifter
00	CRC	O	Internally generated end fill bit
20	SNORM	I	Least significant bit
30	DNORM	I	Least significant bit
60	SMULI	O	ALU0
70	SMULT	O	ALU0
80	SDIVIN	O	Internally generated end fill bit
90	SDIVIS	O	Internally generated end fill bit
A0	SDIVI	O	Internally generated end fill bit
B0	UDIVIS	O	Internally generated end fill bit
C0	UDIVI	O	Internally generated end fill bit
D0	UMULI	O	Internal input
E0	SDIVT	O	Internally generated end fill bit
F0	UDIVIT	O	Internally generated end fill bit
7F	BCDBIN	I	Least significant bit
DF	BINEX3	O	Shifted input to MQ register

Other Arithmetic Instructions

The 'ACT8832 supports two immediate arithmetic operations. ADDI and SUBI (Group 3) add or subtract a constant between the values of 0 and 15 from an operand on the S bus. The constant value is specified in bits A3-A0.

Twelve Group 4 instructions support serial division and multiplication. Signed, unsigned and mixed multiplication are implemented using three instructions: SMULI, which performs a signed times unsigned iteration; SMULT, which provides negative weighting of the sign bit of a negative multiplier in signed multiplication; and UMULI, which performs an unsigned multiplication iteration. Algorithms using these instructions are given in Tables 18, 19, and 20. These include: signed multiplication, which performs a two's complement multiplication; unsigned multiplication, which produces an unsigned times unsigned product; and mixed multiplication which multiplies a signed multiplicand by an unsigned multiplier to produce a signed result.

Table 18. Signed Multiplication Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT Y PORT
E4	LOADMQ	1	Multiplier	—	Multiplier
60	SMULI	N-1†	Accumulator	Multiplicand	Partial product
70	SMULT	1	Accumulator	Multiplicand	Product (MSH)‡

Table 19. Unsigned Multiplication Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT Y PORT
E4	LOADMQ	1	Multiplier	—	Multiplier
D0	UMULI	N-1†	Accumulator	Multiplicand	Partial product
D0	UMULI	1	Accumulator	Multiplicand	Product (MSH)‡

Table 20. Mixed Multiplication Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT Y PORT
E4	LOADMQ	1	Multiplier	—	Multiplier
60	SMULI	N-1†	Accumulator	Multiplicand	Partial product
60	SMULI	1	Accumulator	Multiplicand	Product (MSH)‡

†N = 8 for quad 8-bit mode, 16 for dual 16-bit mode, 32 for 32-bit mode.
‡The least significant half of the product is in the MQ register.

Instructions that support division include start, iterate and terminate instructions for unsigned division routines (UDIVIS, UDIVI and UDIVIT); initialize, start, iterate and terminate instructions for signed division routines (SDIVIN, SDIVIS, SDIVI and SDIVIT); and correction instructions for these routines (DIVRF and SDIVQF). A Group 5 instruction, SDIVO, is available for optional overflow testing. Algorithms for signed and unsigned division are given in Tables 21 and 22. These use a nonrestoring technique to divide a 16 N-bit integer dividend by an 8 N-bit integer divisor to produce an 8 N-bit integer quotient and remainder, where N = 1 for quad 8-bit mode, N = 2 for dual 16-bit mode, and N = 4 for 32-bit mode.

Table 21. Signed Division Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT Y PORT
E4	LOADMQ	1	Dividend (LSH)	—	Dividend (LSH)
80	SDIVIN	1	Dividend (MSH)	Divisor	Remainder (N)
AF	SDIVO	1	Remainder (N)	Divisor	Overflow Test Result
90	SDIVIS	1	Remainder (N)	Divisor	Remainder (N)
A0	SDIVI	N-2†	Remainder (N)	Divisor	Remainder (N)
E0	SDIVIT	1	Remainder (N)	Divisor	Remainder§
40	DIVRF	1	Remainder‡	Divisor	Remainder¶
50	SDIVQF	1	MQ register	Divisor	Quotient#

†N = 8 for quad 8-bit mode, 16 for dual 16-bit mode, 32 for 32-bit mode.
‡The least significant half of the product is in the MQ register.
§Unfixed
¶Fixed (corrected)
#The quotient is stored in the MQ register. Remainder can be output at the Y port or stored in the register file accumulator.

Table 22. Unsigned Division Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT Y PORT
E4	LOADMQ	1	Dividend (LSH)	—	Dividend (LSH)
B0	UDIVIS	1	Dividend (MSH)	Divisor	Remainder (N)
C0	UDIVI	N-1†	Remainder (N)	Divisor	Remainder (N)
F0	UDIVIT	1	Remainder (N)	Divisor	Remainder‡
40	DIVRF	1	Remainder§	Divisor	Remainder§

†N = 8 in quad 8-bit mode, 16 in dual 16-bit mode, 32 in 32-bit mode
‡Unfixed
§Fixed (corrected)

Data Conversion Instructions

Conversion of binary data to one's and two's complement can be implemented using the INCNR instruction (Group 1). SMTC (Group 3) permits conversion from two's complement representation to sign magnitude representation, or vice versa. Two's complement numbers can be converted to their positive value, using ABS (Group 3).

SNORM and DNORM (Group 4) provide for normalization of signed, single- and double-precision data. The operand is placed in the MQ register and shifted toward the most significant bit until the two most significant bits are of opposite value. Zeroes are shifted into the least significant bit, provided SIO is high or floating. (A low on SIO will shift a one into the least significant bit.) SNORM allows the number of shifts to be counted and stored in one of the register files to provide the exponent.

Data stored in binary-coded decimal form can be converted to binary using BCDBIN (Group 5). A routine for this conversion, given in Table 23, allows the user to convert an N-digit BCD number to a 4N-bit binary number in 4N+8 clock cycles.

Table 23. BCD to Binary Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT DESTINATION
E4	LOADMQ	1	BCD operand	—	MQ reg.
D2	SUBR/MQSLC	1	Accumulator	Accumulator	Accumulator/MQ reg.
D2	SUBR/MQSLC	1	Mask reg.	Mask reg.	Mask reg/MQ reg.
D1	MQSLC	2	Don't care	Don't care	MQ reg.
68	ADDI (15)	1	Accumulator	Decimal 15	Mask reg.
REPEAT N-1 TIMES†					
DA	AND/MQSLC	1	MQ reg.	Mask reg.	Interim reg/MQ reg.
D1	ADD/MQSLC	1	Accumulator	Interim reg.	Interim reg/MQ reg.
7F	BCDBIN	1	Interim reg.	Interim res.	Accumulator/MQ reg.
7F	BCDBIN	1	Accumulator	Interim reg.	Accumulator/MQ reg.
END REPEAT					
FA	AND	1	MQ reg.	Mask reg.	Interim reg.
D1	ADD MQSLC	1	Accumulator	Interim reg.	Accumulator

†N = Number of BCD digits

BINEX3, EX3BC, and EX3C assist binary to excess-3 conversion. Using BINEX3, an N-bit binary number can be converted to an N/4- digit excess-3 number. For an algorithm, see Table 24.

Table 24. BCD to Binary Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT DESTINATION
E4	LOADMQ	1	Binary number	—	MQ reg.
D2	SUBR	1	Accumulator	Accumulator	Accumulator
D2	SET1 (33)16	1	Accumulator	Mask (33)16	Accumulator
REPEAT N TIMES†					
DF	BINEX3	1	Accumulator	Accumulator	Accumulator/MQ reg
9F	EX3C	1	Accumulator	Internal data	Accumulator
END REPEAT					

†N = Number of bits in binary number

Bit and Byte Instructions

Four Group 3 instructions allow the user to test or set selected bits within a byte. SET1 and SET0 force selected bits of a selected byte (or bytes) to one and zero, respectively. TB1 and TB0 test selected bits of a selected byte (or bytes) for ones and zeros. The bits to be set or tested are specified by an 8-bit mask formed by the concatentation of register file address inputs C3-C0 and A3-A0. The register file addressed by B5-B0 is used as the destination operand for the set bit instructions. Register writes are inhibited for test bit instructions. Bytes to be operated on are selected by forcing SIOn low, where n represents the byte position and 0 represents the least significant byte. A high on the zero output pin signifies that the test data matches the mask; a low on the zero output indicates that the test has failed.

Individual bytes of data can also be manipulated using eight Group 3 byte arithmetic/logic instructions. Bytes can be added, subtracted, incremented, ORed, ANDed and exclusive ORed. Like the bit instructions, bytes are selected by forcing SIOn low, but multiple bytes can be operated on only if they are adjacent to one another; at least one byte must be nonselected.

Other Instructions

SEL (Group 4) selects one of the ALU's two operands, S or R, depending on the state of the SSF pin. This instruction could be used in sort routines to select the larger or smaller of two operands by performing a subtraction and sending the status result to SSF. CRC (Group 4) is designed to verify serial binary data that has been transmitted over a channel using a cyclic redundancy check code. An algorithm using this instruction is given in Table 25.

Table 25. CRC Algorithm

OP CODE	MNEMONIC	CLOCK CYCLES	INPUT S PORT	INPUT R PORT	OUTPUT DESTINATION
E4	LOADMQ	1	Vector c'(x)†	—	MQ reg.
F6	INCR	1	—	Polynomial g(x)	Poly reg.
F2	SUBR	1	Accumulator	Accumulator	Accumulator
REPEAT n/8N TIMES†					
00	CRC	1	Accumulator	Poly reg.	Accumulator
E4	LOADMQ	1	Vector c'(x)†	—	MQ reg.
END REPEAT					

†N = Number of bits in binary number
n = Length of the code vector

CLR forces the ALU output to zero and clears the internal BCD flip-flops used in excess-3 BCD operations. NOP forces the ALU output to zero, but does not affect the flip-flops.

Configuration Options

The 'ACT8832 can be configured to operate in 8-bit, 16-bit, or 32-bit modes, depending on the setting of the configuration mode selects (CF2-CF0). Table 11 shows the control inputs for the four operating modes. Selecting an operating configuration other than 32-bit mode affects ALU operation and status generation in several ways, depending on the mode selected.

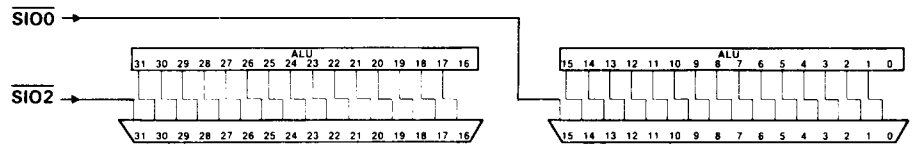
Masked 32-Bit Operation

Masked 32-bit operation is selected to reset to zero the 20 most significant bits of the R Mux input. The 12 least significant bits are unaffected by the mask. Only Group 1 and Group 2 instructions can be used in this operating configuration. Status generation is similar to unmasked 32-bit operating mode.

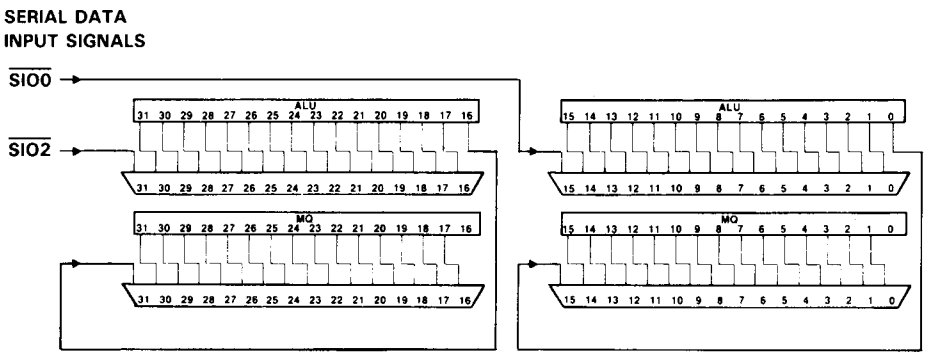
Shift Instructions

Shift instructions operate similarly in 8-bit, 16-bit, and 32-bit modes. The serial I/O (SIO3'-SIO0') pins are used to select end-fill bits or to shift bits in or out, depending on the operation being performed. Table 12 shows the $\overline{SIO}$ signals associated with each byte or word in the different modes, and Table 17 indicates the specific function performed by the $\overline{SIO}$ pins during shift, multiply, and divide operations.

Figures 9 and 10 present examples of logical right shifts in 16-bit and 8-bit configurations.



Single Precision Logical Right Single Shift, 16-Bit Configuration



Double Precision Logical Right Single Shift, 16-Bit Configuration

Figure 9. Shift Examples, 16-Bit Configuration

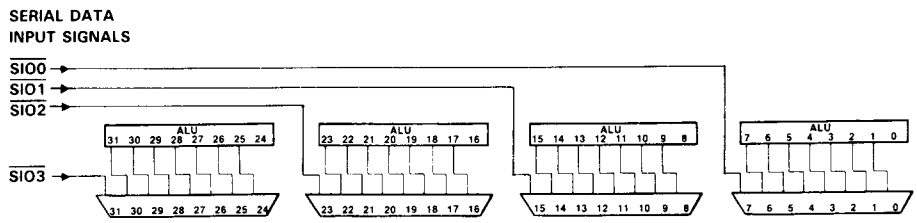
Bit and Byte Instructions

The 'ACT8832 performs bit operations similarly in 8-bit, 16-bit, and 32-bit modes. Masks are loaded into the R MUX on the A3-A0 and C3-C0 address inputs, and the bytes to be masked are selected by pulling their $\overline{SIO}$' inputs low. Instructions which set, reset, or test bits are explained later

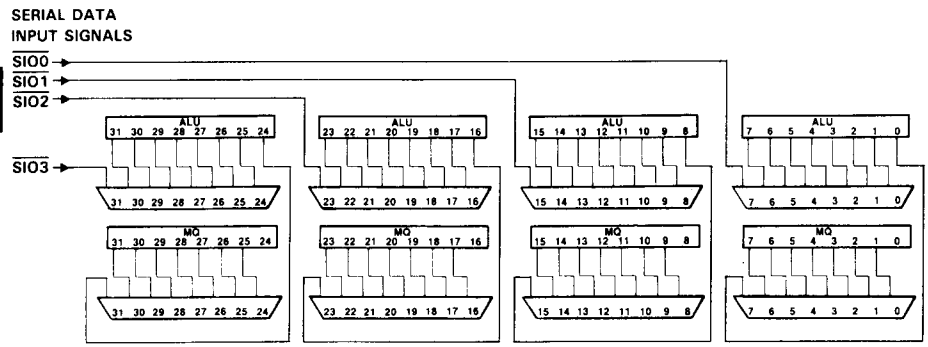
Byte operations should be performed in 32-bit mode to get the necessary status outputs. While byte overflow signals are provided for all four bytes (BYOF3-BYOF0), the other status signals (C, N, Z) are output only for the word selected with the configuration control signals (CF2-CF0).

Status Selection

Status results (C, N, Z, and overflow) are internally generated for all words in all modes, but only the overflow results (BYOF3-BYOF0) are available for all four bytes in 8-bit mode or for both words in 16-bit mode. If a specific application requires that the four status results are read for two or four words, it is possible to toggle the configuration



Single-Precision Logical Right Shift, 8-Bit Configuration



Double-Precision Logical Right Shift, 8-Bit Configuration

Figure 10. Shift Examples, 8-Bit Configuration

control signals (CF2-CF0) within the same clock cycle and read the additional status results. This assumes that the necessary external hardware is provided to toggle CF2-CF0 and collect the status for the individual words before the next clock signal is input.

Instruction Set

The 'ACT8832 instruction set is presented in alphabetical order on the following pages. The discussion of each instruction includes a functional description, list of possible operands, data flow diagram, and notes on status and control bits affected by the instruction. Microcoded examples are also shown.

Mnemonics and opcodes for instructions are given at the top of each page. Opcodes for instructions in Groups 1 and 2 are four bits long and are combined into eight-bit instructions which select combinations of arithmetic, logical, and shift operations. Opcodes for the other instruction groups are all eight bits long.

An asterisk in the left side of the opcode box for a Group 1 instruction indicates that a Group 2 opcode is needed to complete the instruction. An asterisk in the right side of a box indicates that a Group 1 opcode is required to combine with the Group 2 opcode in the left side of the box.

FUNCTION

Computes the absolute value of two's complement data on the S bus.

DESCRIPTION

Two's complement data on the S bus is converted to its absolute value. The carry must be set to one by the user for proper conversion. ABS causes S' + Cn to be computed; the state of the sign bit determines whether S or S' + Cn will be selected as the result. SSF is used to transmit the sign of S.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Should be programmed high for proper conversion.

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB (input) = 1
OVR = 1 if input of most significant byte is 80 (Hex) and inputs (if any) in all other bytes are 00 (Hex).
C = 1 if S = 0

EXAMPLES (assumes a 32-bit configuration)

Convert the two's complement number in register 1 to its positive value and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$ $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0100 1000	XX XXXX	00 0001	X 00	00 0100	0	0000	10	X	X	XXXX	0	1	110

Example 1: Assume register file 1 holds F6D81340 (Hex):

Source: 1111 0110 1101 1000 0001 0011 0100 0000 S ← RF(1)

Destination: 0000 1001 0010 0111 1110 1100 1100 0000 RF(4) ← S + Cn

Example 2: Assume register file 1 holds 09D527C0 (Hex):

Source: 0000 1001 1101 0101 0010 0111 1100 0000 S ← RF(1)

Destination: 0000 1001 1101 0101 0010 0111 1100 0000 RF(4) ← S

FUNCTION

Adds data on the R and S buses to the carry-in.

DESCRIPTION

Data on the R and S buses is added with carry. The sum appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Increments sum if set to one.

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 1 if signed arithmetic overflow
C = 1 if carry-out = 1

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLES (assumes a 32-bit configuration)

Add data in register 1 to data on the DB bus with carry-in and pass the result to the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	WE3- WE0	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- OEY0	$\overline{OES}$	Cn	CF2- CF0
1110 0001	00 0001	XX XXXX	0 10	XX XXXX	0	1111	10	X	X	XXXX	0	0	110

Assume register file 1 holds 0802C618 (Hex and DB bus holds 1E007530 (Hex):

Source 0000 1000 0000 0010 1100 0110 0001 1000 R ← RF(1)

Source 0001 1110 0000 0000 0111 0101 0011 0000 S ← DB bus

Destination 0010 0110 0000 0011 0011 1011 0100 1000 MQ register ← R + S + Cn

FUNCTION

Adds four-bit immediate data on A3-A0 with carry to S-bus data.

DESCRIPTION

Immediate data in the range 0 to 15, supplied by the user at A3-A0, is added with carry to S.

Available R Bus Source Operands (Constant)

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: Mask
No	Yes	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Increments sum if set to one.

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 1 if signed arithmetic overflow
C = 1 if carry-out = 1

EXAMPLES (assumes a 32-bit configuration)

Add the valule 12 to data on the DB bus with carry-in and store the result in register file 1.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel $\overline{EA}$ EB1- EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$ $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0110 1000	00 1100	XX XXXX	X 10	00 0001	0	0000	10	X	X	XXXX	0	0	110

Assume bits A5-A0 hold 0C (Hex) and DB bus holds 24000100 (Hex):

Source 0000 0000 0000 0000 0000 0000 0000 1100 R ← A5-A0

Source 0010 0100 0000 0000 0000 0001 0000 0000 S ← DB bus

Destination 0010 0100 0000 0000 0000 0001 0000 1100 RF(1) ← R + S + Cn

FUNCTION

Evaluates the logical expression R AND S.

DESCRIPTION

Data on the R bus is ANDed with data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals†

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 0
C	= 0

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLES (assumes a 32-bit configuration)

Logically AND the contents of register 3 and register 5 and store the result in register 5.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$ $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1010	00 0011	00 0101	0 00	00 0101	0	0000	10	X	X	XXXX	0	X	110

Assume register file 3 holds F617D840 (Hex) and register file 5 holds 15F6D842 (Hex):

Source 1111 0110 0001 0111 1101 1000 0100 0000 R ← RF(3)

Source 0001 0101 1111 0110 1101 1000 0100 0010 S ← RF(5)

Destination 0001 0100 0001 0110 1101 1000 0100 0000 RF(5) ← R AND S

FUNCTION

Computes the logical expression S AND NOT R.

DESCRIPTION

The logical expression S AND NOT R is computed. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals†

ZERO	= 1 if result = 0
N	= 0
OVR	= 0
C	= 0

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Invert the contents of register 3, logically AND the result with data in register 5 and store the result in register 10.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1110	00 0011	00 0101	0 00	00 1010	0	0000	10	X	X	XXXX	0	X	110

Assume register file 3 holds 15F6D840 (Hex) and register file 5 hold F617D842 (Hex):

Source `0001 0101 1111 0110 1101 1000 0100 0000` R ← RF(3)

Source `1111 0110 0001 0111 1101 1000 0100 0010` S ← RF(5)

Destination `1110 0010 0000 0001 0000 0000 0000 0010` RF(10) ← $\overline{R}$ AND S

FUNCTION

Adds S with carry-in to a selected byte or selected adjacent bytes of R.

DESCRIPTION

$\overline{SIO3}$-$\overline{SIO0}$ are used to select bytes of R to be added to the corresponding bytes of S. A byte of R with $\overline{SIO}$ programmed low is selected for the computation of R + S + Cn. If the $\overline{SIO}$ signal for a byte of R is left high, the corresponding byte of S is passed unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	Yes	Propagates through nonselected bytes; increments selected byte(s) if programmed high.

Status Signals

ZERO	=	1 if result (selected bytes) = 0
N	=	0
OVR	=	1 if signed arithmetic overflow (selected bytes)
C	=	1 if carry-out (most significant selected byte) = 1

EXAMPLE (assumes a 32-bit configuration)

Add bytes 1 and 2 of register 3 with carry to the contents of register 1 and store the result in register 11.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0100 1000	00 0011	00 0001	0 00	00 1011	0	0000	10	X	X	XXXX	0	1	110	1001	0000

Assume register file 3 holds 2C018181 (Hex) and register file 1 holds 7A8FBE3E (Hex):

Source	0010 1100 0000 0001 1000 0001 1000 0001	Rn ← RF(3)n
Source	0111 1010 1000 1111 1011 1110 0011 1110	Sn ← RF(1)n
ALU	1010 0110 1001 0001 0100 0000 1100 0000	Fn ← Rn + Sn + Cn
Destination	0111 1010 1001 0001 0100 1111 0011 1110	RF(11)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 11 gets F if byte selected, S if byte not selected.

FUNCTION

Evaluates the logical AND of selected bytes of R-bus and S-bus data.

DESCRIPTION

Bytes with their corresponding $\overline{SIO}$ signals programmed low compute R AND S. Bytes with $\overline{SIO}$ signals programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Forced low
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	No	Inactive

Status Signals

ZERO = 1 if result (selected bytes) = 0
N = 0
OVR = 0
C = 0

EXAMPLE (assumes a 32-bit configuration)

Logically AND bytes 1 and 2 of register 3 with input on the DB bus; store the result in register 3.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1110 1000	00 0011	XX XXXX	0 10	00 0011	0	0000	10	X	X	XXXX	0	X	110	1001	0000

Assume register file 3 holds 398FBEBE (Hex) and input on the DB port is 4290BFBF (Hex):

Source 0011 1001 1000 1111 1011 1110 1011 1110 Rn ← RF(3)n

Source 0100 0010 1001 0000 1011 1111 1011 1111 Sn ← DBn

Destination 0100 0010 1000 0000 1011 1110 1011 1111 RF(3)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 3 gets F if byte selected, S if byte not selected.

FUNCTION

Converts a BCD number to binary.

DESCRIPTION

This instruction allows the user to convert an N-digit BCD number to a 4N-bit binary number in 4(N-1) plus 8 clocks. The instruction sums the R and S buses with carry.

A one-bit arithmetic left shift is performed on the ALU output. A zero is filled into bit 0 of the least significant byte unless $\overline{SIOO}$ is set low, which would force bit 0 to one. Bit 7 of the most significant byte is dropped.

Simultaneously, the contents of the MQ register are rotated one bit to the left. Bit 7 of the most significant byte is rotated to bit 0 of the least significant byte.

Recommended R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	No	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	If high or floating, fills a zero in LSB of ALU shifter; if low, fills a one in LSB of ALU shifter.
$\overline{SIO1}$	No	Inactive in 32-bit configuration. Used in other configurations to select endfill in LSBs.
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed low for proper conversion.

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 1 if signed arithmetic overflow
C = 1 if carry-out = 1

ALGORITHM

The following code converts an N-digit BCD number to a 4N-bit binary number in 4(N-1) plus 8 clocks. This is one possible user generated algorithm. It employs the standard conversion formula for a BCD number (shown here for 32 bits):

$$ABCD = [(A \times 10 + B) \times 10 + C] \times 10 + D.$$

The conversion begins with the most significant BCD digit. Addition is performed in radix 2.

PSEUDOCODE

LOADMQ	NUM	Load MQ with BCD number.
SUB	ACC, ACC, SLCMQ	Clear accumulator; Circular left shift MQ.
SUB	MSK, MSK, SLCMQ	Clear mask register; Circular left shift MQ.
SLCMQ		Circular left shift MQ.
SLCMQ		Circular left shift MQ.
ADDI	ACC, MSK, 15	Store 15 in mask register.

Repeat N-1 times:

(N = number of BCD digits)

AND	MQ, MSK, R1, SLCMQ	Extract one digit; Circular left shift MQ.
ADD	ACC, R1, R1, SLCMQ	Add extracted digit to accumulator, and store result in R1; Circular left shift MQ.
BCDBIN	R1, R1, ACC	Perform BCDBIN instruction, and store result in accumulator [4 × (ACC + 4 × digit)]; Circular left shift MQ.
BCDBIN	ACC, R1, ACC	Perform BCDBIN instruction, and store result in accumulator [10 × (ACC + 10 × digit)]; Circular left shift MQ.

(END REPEAT)

AND	MQ MSK, R1	Fetch last digit.
ADD	ACC, R1, ACC	Add in last digit and store result in accumulator.

FUNCTION

S' + Cn for selected bytes of S.

DESCRIPTION

Bytes with $\overline{SIO0}$ programmed low compute S' + Cn. Bytes with $\overline{SIO0}$ programmed high pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	Yes	Propagates through nonselected bytes; increments selected byte(s) if programmed high.

Status Signals

ZERO = 1 if result (selected bytes) = 0
N = 0
OVR = 1 if signed arithmetic overflow (selected bytes)
C = 1 if carry-out (most significant selected byte) = 1

EXAMPLE (assumes a 32-bit configuration)

Invert bytes 0 and 1 of register 3 and add them to the carry (bytes 2 and 3 are not changed). Store the result in register 3.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1100 1000	XX XXXX	00 0001	X 00	00 0011	0	0000	10	X	X	XXXX	0	1	110	1100	0000

Assume register file 3 holds A3018181 (Hex):

Source: 1010 0011 0000 0001 1000 0001 1000 0001 — Sn ← RF(3)n

ALU: 0101 1100 1111 1110 0111 1110 0111 1111 — Fn ← S'n + Cn

Destination: 1010 0011 0000 0001 0111 1110 0111 1111 — RF(3)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 3 gets F if byte selected, S if byte not selected.

FUNCTION

Increments selected bytes of S if the carry is set.

DESCRIPTION

Bytes with SIO' inputs programmed low compute S + Cn. Bytes with $\overline{SIO}$ inputs programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonsselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	Yes	Propagates through nonselected bytes; increments selected byte(s) if programmed high.

Status Signals

ZERO = 1 if result (selected bytes) = 0
N = 0
OVR = 1 if signed arithmetic overflow (selected bytes)
C = 1 if carry-out (most significant selected byte) = 1

EXAMPLE (assumes a 32-bit configuration)

Add bytes 1 and 2 of register 7 to the carry (bytes 0 and 3 are not changed). Store the result in register 2.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1011 1000	XX XXXX	00 0111	X 00	00 0010	0	0000	10	X	X	XXXX	0	1	110	1100	0000

Assume register file 7 holds 408FBEBE (Hex):

Source 0100 0000 1000 1111 1011 1110 1011 1110 Sn ← RF(7)n

ALU 0100 0000 1000 1111 1011 1111 1011 1110 Fn ← Sn + Cn

Destination 0100 0000 1000 1111 1011 1111 1011 1110 RF(2)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 11 gets F if byte selected, S if byte not selected.

FUNCTION

Converts a binary number to excess-3 representation.

DESCRIPTION

This instruction converts an N-digit binary number to a N/4 digit excess-3 number representation in 2N + 3 clocks. The data on the R and S buses are added to the carry-in, which contains the most significant bit of the MQ register. The contents of the MQ register are rotated one bit to the left. The most significant bit is shifted out and passed to the least significant bit position. Depending on the configuration selected, this shift may be within the same byte or from the most significant byte to the least significant byte.

Recommended R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	No	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Holds MSB of MQ register.

Status Signals

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

ALGORITHM

The following code converts an N-digit binary number to a N/4 digit excess-3 number in 2N+3 clocks. It employs the standard conversion formula for a binary number:

$$a_n 2^n + a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \ldots + a_0 =$$
$$\{[(2a_n + a_{n-1}) \times 2 + a_{n-1}] \times 2 + \ldots + a_0\} \times 2 + a^o.$$

The conversion begins with the most significant bit. Addition during the BINEX3 instruction is performed in radix 10 (excess-3).

LOADMQ NUM	Load MQ with binary number.
SUB ACC, ACC, ACC	Clear accumulator;
SET1 ACC, 33 (Hex)	Store 33 (Hex) in all bytes of accumulator.

Repeat N times:
(N = number of bits in binary number)

BINEX3 ACC, ACC, ACC	Double accumulator and add in most significant bit of MQ register. Circular left shift MQ.
EX3C ACC	Perform excess-3 correction.

(END REPEAT)

FUNCTION

Evaluates R OR S of selected bytes.

DESCRIPTION

Bytes with $\overline{SIO}$ inputs programmed low evaluate R OR S. Bytes with $\overline{SIO}$ inputs programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	No	Inactive

Status Signals

ZERO	=	1 if result (selected bytes) = 0
N	=	0
OVR	=	0
C	=	0

EXAMPLE (assumes a 32-bit configuration)

Logically OR bytes 1 and 2 of register 12 with bytes 1 and 2 on the DB bus. Concatenate with DB bytes 0 and 3, storing the result in register 12.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1111 1000	00 1100	XX XXXX	0 10	00 1100	0	0000	10	X	X	XXXX	0	X	110	1001	0000

Assume register file 12 holds 578FBEBE (Hex) and the DB bus holds 1C90BEBE (Hex):

Source	0101 0111 1000 1111 1011 1110 1011 1110	Rn ← RF(12)n
Source	0001 1100 1001 0000 1011 1110 1011 1100	Sn ← DBn
Destination	0001 1100 1001 1111 1011 1110 1011 1110	RF(12)n ← Fn or Sn†

†F = ALU result
n = nth package
Register file 12 gets F if byte selected, S if byte not selected.

FUNCTION

Subtracts R from S in selected bytes.

DESCRIPTION

Bytes with $\overline{SIO}$ inputs programmed low compute R' + S + Cn. Bytes with $\overline{SIO}$ inputs programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	Yes	Propagates through nonselected bytes; should be set high for two's complement subtraction.

Status Signals

ZERO = 1 if result (selected bytes) = 0
N = 0
OVR = 1 if signed arithmetic overflow (selected bytes)
C = 1 if carry-out (most significant selected byte) = 1

EXAMPLE (assumes a 32-bit configuration)

Subtract bytes 1 and 2 of register 1 with carry from bytes 1 and 2 of register 3. Concatenate with bytes 0 and 3 of register 3, storing the result in register 11.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1010 1000	00 0001	00 0011	0 00	00 1011	0	0000	10	X	X	XXXX	0	1	110	1001	0000

Assume register file 1 holds 091B5858 (Hex) and register file 3 holds 703A9898 (Hex):

Source: 0000 1001 0001 1011 0101 1000 0101 1000 — Rn ← RF(1)n

Source: 0111 0000 0011 1010 1001 1000 1001 1000 — Sn ← RF(3)n

ALU: 0110 0111 0001 1111 0100 0000 0100 0000 — Fn ← R'n + Sn + Cn

Destination: 0111 0000 0001 1111 0100 0000 1001 1000 — RF(11)n ← Fn or Sn†

†F = ALU result
n = nth package
Register file 11 gets F if byte selected, S if byte not selected.

FUNCTION

Subtracts S from R in selected bytes.

DESCRIPTION

Bytes with $\overline{SIO}$ inputs programmed low compute R + S' + Cn. Bytes with $\overline{SIO}$ inputs programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	Yes	Propagates through nonselected bytes; should be set high for two's complement subtraction.

Status Signals

ZERO	=	1 if result (selected bytes) = 0
N	=	0
OVR	=	1 if signed arithmetic overflow (selected bytes)
C	=	1 if carry-out (most significant selected byte) = 1

EXAMPLE (assumes a 32-bit configuration)

Subtract bytes 1 and 2 of register 3 with carry from bytes 1 and 2 of register 1. Concatenate with bytes 0 and 3 of register 3, storing the result in register 11.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1001 1000	00 0001	00 0011	0 00	00 1011	0	0000	10	X	X	XXXX	0	1	110	1001	0000

Assume register file 1 holds 5288B8B8 (Hex) and register file 3 holds 143A9898 (Hex):

Source	0101 0010 1000 1000 1011 1000 1011 1000	Rn ← RF(1)n
Source	0001 0100 0011 1010 1001 1000 1001 1000	Sn ← RF(3)n
ALU	0011 1110 0100 1110 0010 0000 0010 0000	Fn ← Rn + S'n + Cn
Destination	0101 0010 0100 1110 0010 0000 1011 1000	RF(11)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 11 gets F if byte selected, S if byte not selected.

FUNCTION

Evaluates R exclusive OR S in selected bytes.

DESCRIPTION

Bytes with $\overline{SIO}$ inputs programmed low evaluate R exclusive OR S. Bytes with $\overline{SIO}$ inputs programmed high, pass S unaltered. Multiple bytes can be selected only if they are adjacent to one another. At least one byte must be nonselected.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	No	Inactive

Status Signals

ZERO	=	1 if result (selected bytes) = 0
N	=	0
OVR	=	0
C	=	0

EXAMPLE (assumes a 32-bit configuration)

Exclusive OR bytes 1 and 2 of register 6 with bytes 1 and 2 on the DB bus; concatenate the result with DB bytes 0 and 3, storing the result in register 10.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1101 1000	00 0110	XX XXXX	0 10	00 1010	0	0000	10	X	X	XXXX	0	1	110	1001	0000

Assume register file 6 holds 938FBEBE (Hex) and the DB bus holds 4190BEBE (Hex):

Source: 1001 0011 1000 1111 1011 1110 1011 1110 — Rn ← RF(6)n

Source: 0100 0001 1001 0000 1011 1110 1011 1110 — Sn ← DBn

Destination: 0100 0001 0001 1111 0000 0000 1011 1110 — RF(10)n ← Fn or Sn†

†F = ALU result
n = nth package
Register file 10 gets F if byte selected, S if byte not selected.

FUNCTION

Forces ALU output to zero and clears the BCD flip-flops.

DESCRIPTION

ALU output is forced to zero and the BCD flip-flops are cleared.

†This instruction may also be coded with the following opcodes:
[2] [F], [3] [F], [4] [F], [6] [F], [B] [F], [C] [F], [E] [F]

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
No	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Status Signals

ZERO = 1
N = 0
OVR = 0
Cn = 0

FUNCTION

Evaluates R exclusive OR S for use with cyclic redundancy check codes.

DESCRIPTION

Data on the R bus is exclusive ORed with data on the S bus. If MQ0 XNORed with S0 is zero (MQ0 is the LSB of the MQ register and S0 is the LSB of S-bus data), the result is sent to the ALU shifter. Otherwise, data on the S bus is sent to the ALU shifter.

A right shift is performed; the MSB is filled with R0 (MQ0 XOR S0), where R0 is the LSB of R-bus data. A circular right shift is performed on MQ data.

Recommended R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	No	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
Right	Right

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals

ZERO = 1 if result = 0
N = 0
OVR = 0
Cn = 0

CYCLIC REDUNDANCY CHARACTER CHECK

DESCRIPTION

Serial binary data transmitted over a channel is susceptible to error bursts. These bursts may be detected and corrected by standard encoding methods such as cyclic redundancy check codes, fire codes, or computer generated codes. These codes all divide the message vector by a generator polynomial to produce a remainder that contains parity information about the message vector.

If a message vector of m bits, a(x), is divided by a generator polynomial, g(x), of order k-1, a k bit remainder, r(x), is formed. The code vector, c(x), consisting of m(x) and r(x) of length n = m+k is transmitted down the channel. The receiver divides the received vector by g(x).

After m divide iterations, r(x) will be regenerated only if there is no error in the message bits. After k more iterations, the result will be zero if and only if no error has occurred in either the message or the remainder.

ALGORITHM

An algorithm for a cyclic redundancy character check, using the 'ACT8832 as a receiver, is given below:

LOADMQ VEC(X)	Load MQ with first 32 message bits of received vector c'(x).
LOAD POLY	Load register with polynomial g(x).
CLEAR SUM	Clear register acting as accumulator.
REPEAT (n/32) TIMES:	
SUM = SUM CRC POLY	Perform CRC instruction where R Bus = POLY S Bus = SUM Store result in SUM.
LOADMQ VEC(X)	Load MQ with next 32 message bits of received vector c'(x).
(END REPEAT)	

SUM now contains the remainder [r'(x)] of c'(x). A syndrome generation routine may be called next, if required.

Note that the most significant bit of

$$g(x) = (g_{k-1})(x^{k-1}) + (g_{k-2})(x^{k-2}) + \ldots(g_0)(x^0)$$

is implied and that POLY(0) is set to zero if the length of g(x) requires fewer bits than are in the machine word width.

FUNCTION

Corrects the remainder of nonrestoring division routine if correction is required.

DESCRIPTION

DIVRF tests the result of the final step in nonrestoring division iteration: SDIVIT (for signed division) or UDIVIT (for unsigned division). An error in the remainder results when it is nonzero and the signs of the remainder and the dividend are different.

The R bus must be loaded with the divisor and the S bus with the most significant half of the previous result. The least significant half is in the MQ register. The Y bus result must be stored in the register file for use during the subsequent SDIVQF instruction.

DIVRF tests to determine whether a fix is required and evaluates:

$Y \leftarrow S + R' + 1$ if a fix is necessary

$Y \leftarrow S + R + 0$ if a fix is unnecessary

Overflow is reported to OVR at the end of the division routine (after SDIVQF).

Recommended R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	No	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Should be programmed high

Status Signals

ZERO = 1 if remainder = 0
N = 0
OVR = 0
Cn = 1 if carry-out = 1

FUNCTION

Tests the two most significant bits of a double precision number. If they are the same, shifts the number to the left.

DESCRIPTION

This instruction is used to normalize a two's complement, double precision number by shifting the number one bit to the left and filling a zero into the LSB unless $\overline{SIOO}$ is low. The S bus holds the most significant half; the MQ register holds the least significant half.

Normalization is complete when overflow occurs. The shift is inhibited whenever normalization is attempted on a number already normalized.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Recommended S Bus Source Operands (MSH)

RF (B5-B0)	DB-Port	MQ Register
Yes	No	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations (conditional)

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	When low, selects a one end-fill bit in LSB
$\overline{SIO1}$	No	Passes internally generated end-fill bits
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 1 if MSB XOR 2nd MSB = 1
Cn = 0

EXAMPLE (assumes a 32-bit configuration)

Normalize a double-precision number.

(This example assumes that the MSH of the number to be normalized is in register 3 and the LSH is in the MQ register. The zero on the OVR pin at the end of the instruction cycle indicates that normalization is not complete and the instruction should be repeated).

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0011 0000	XX XXXX	00 0011	X 00	00 0011	0	0000	10	X	X	XXXX	0	X	110

Assume register file 3 holds FA75D84E (Hex) and MQ register holds 37F6D843 (Hex):

Source 1111 1010 0111 0101 1101 1000 0100 1110 ALU shifter ← RF(3)

Source 0011 0111 1111 0110 1101 1000 0100 0011 MQ shifter ← MQ register

Destination 1111 0100 1110 1011 1011 0000 1001 1101 8RF(3) ← Result (MSH)

Destination 0110 1111 1110 1101 1011 0000 1000 0110 MQ register ← Result (LSH)

0 OVR ← 0†

†Normalization not complete at the end of this instruction cycle.

FUNCTION

Output contents of the divide/BCD flip-flops.

DESCRIPTION

The contents of the divide/BCD flip-flops are passed through the MQ register to the Y output Imultiplexer.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
No	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
No	No	Yes

Shift Operations

ALU	MQ
None	None

Status Signals

ZERO = 0
N = 0
OVR = 0
Cn = 0

EXAMPLES (assumes a 32-bit configuration)

Dump divide/BCD flip-flops to Y output.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0101 1111	XX XXXX	XX XXXX	X XX	XX XXXX	1	XXXX	XX	X	X	0000	X	X	110

Assume divide/BCD flip-flops contain 2A055470 (Hex):

Source 0010 1010 0000 0101 0101 0100 0111 0000 MQ register ← Divide/BCD flip-flops

Destination 0010 1010 0000 0101 0101 0100 0111 0000 Y output ← MQ register

FUNCTION

Corrects the result of excess-3 addition or subtraction in selected bytes.

DESCRIPTION

This instruction corrects excess-3 additions or subtractions in the byte mode. For correct excess-3 arithmetic, this instruction must follow each add or subtract. The operand must be on the S bus.

Data on the S bus is added to a constant on the R bus determined by the state of the BCD flip flops and previous overflow condition reported on the SSF pin. Bytes with $\overline{SIO}$ inputs programmed low evaluate the correct excess-3 representation. Bytes with $\overline{SIO}$ inputs programmed high or floating, pass S unaltered.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
No	No

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte select
$\overline{SIO1}$	Yes	Byte select
$\overline{SIO2}$	Yes	Byte select
$\overline{SIO3}$	Yes	Byte select
Cn	No	Inactive

Status Signals

ZERO = 0
N = 0
OVR = 1 if arithmetic signed overflow
Cn = 1 if carry-out = 1

EXAMPLE (assumes a 32-bit configuration)

Add two BCD numbers and store the sum in register 3. Assume data comes in on DB bus.

1. Clear accumulator (SUB ACC, ACC)
2. Store 33 (Hex) in all bytes of register (SET1 R2, H/33/)
3. Add 33 (Hex) to selected bytes of first BCD number (BADD DB, R2, R1)
4. Add 33 (Hex) to selected bytes of second BCD number (BADD DB, R2, R3)
5. Add selected bytes of registers 1 and 3 (BADD, R1, R3, R3)
6. Correct the result (EX3BC, R3, R3)

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1111 0010	00 0010	XX XXXX	0 XX	00 0010	0	0000	10	X	X	XXXX	0	1	110	XXXX	XXXX
0000 1000	00 0010	XX XXXX	0 XX	00 0010	0	0000	10	X	X	XXXX	0	X	110	XXXX	XXXX
1000 1000	00 0010	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	0	110	1100	0000
1000 1000	00 0010	XX XXXX	0 10	00 0011	0	0000	10	X	X	XXXX	0	0	110	1100	0000
1000 1000	00 0001	00 0011	0 00	00 0011	0	0000	10	X	X	XXXX	0	0	110	1100	0000
1000 1111	XX XXXX	00 0011	X 00	00 0011	0	0000	10	X	X	XXXX	0	0	110	1100	0000

Assume DB bus holds 51336912 at third instruction and 34867162 at fourth instruction.

1 | 0000 0000 0000 0000 0000 0000 0000 0000 | RF(2) ← 0

2 | 0000 0000 0000 0000 0011 0011 0011 0011 | RF(2) ← 00003333 (Hex)

3 | 0101 0001 0011 0011 1001 1100 0100 0101 | RF(1) ← RF(2) + DB

4 | 0011 0100 1000 0110 1010 0100 1001 0101 | RF(3) ← RF(2) + DB

5 | 0011 0100 1000 0110 0100 0000 1101 1010 | RF(3)n ← RF(1)n + RF(3)n

6 | 0011 0100 1000 0110 0100 0000 0111 0100 | RF(3)n ← Corrected RF(3)n result

FUNCTION

Corrects the result of excess-3 addition or subtraction.

DESCRIPTION

This instruction corrects excess-3 additions or subtractions in the word mode. For correct excess-3 arithmetic, this instruction must follow each add or subtract. The operand must be on the S bus.

Data on the S bus is added to a constant on the R bus determined by the state of the BCD flip-flops and previous overflow condition reported on the SSF pin.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
No	No

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals

ZERO = 0
N = 1 if MSB = 1
OVR = 1 if arithmetic signed overflow
Cn = 1 if carry-out = 1

EXAMPLE (assumes a 32-bit configuration)

Add two BCD numbers and store the sum in register 3. Assume data comes in on DA bus.

1. Clear accumulator (SUB ACC, ACC)
2. Store 33 (Hex) in all bytes of register (SET1 R2, H/33/)
3. Add 33 (Hex) to all bytes of first BCD number (ADD DB, R2, R1)
4. Add 33 (Hex) to all bytes of second BCD number (ADD DB, R2, R3)
5. Add the excess-3 data (ADD, R1, R3, R3)
6. Correct the excess-3 result (EX3C, R3, R3)
7. Subtract the excess-3 bias to go to BCD result.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel $\overline{EA}$	Oprd Sel EB1-EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$-$\overline{WE0}$	SELRF1-SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$-$\overline{OEY0}$	$\overline{OES}$	Cn	CF2-CF0
1111 0010	00 0010	XX XXXX	0	XX	00 0010	0	0000	10	X	X	XXXX	0	1	110
0000 1000	00 0010	XX XXXX	0	XX	00 0010	0	0000	10	X	X	XXXX	0	X	110
1111 0001	00 0010	XX XXXX	0	10	00 0001	0	0000	10	X	X	XXXX	0	0	110
1111 0001	00 0010	XX XXXX	0	10	00 0011	0	0000	10	X	X	XXXX	0	0	110
1111 0001	00 0001	00 0011	0	00	00 0011	0	0000	10	X	X	XXXX	0	0	110
1001 1111	XX XXXX	00 0011	X	00	00 0011	0	0000	10	X	X	XXXX	0	0	110
1111 0010	00 0010	00 0011	0	00	00 0011	0	0000	10	X	X	XXXX	0	0	110

Assume DB bus holds 51336912 at third instruction and 34867162 at fourth instruction.

Results of Instruction Cycles:

1	0000 0000 0000 0000 0000 0000 0000 0000	RF(2) ← 0
2	0011 0011 0011 0011 0011 0011 0011 0011	RF(2) ← 33333333 (Hex)
3	1000 0100 0110 0110 1001 1100 0100 0101	RF(1) ← RF(2) +DB
4	0110 0111 1011 1001 1010 0100 1001 0101	RF(3) ← RF(2) + DB
5	1110 1100 0010 0000 0100 0000 1101 1010	RF(3) ← RF(1) + RF(3)
6	1011 1001 0101 0011 0111 0011 1010 0111	RF(3) ← Corrected RF(3) result
7	1000 0110 0010 0000 0100 0000 0111 0100	RF(3) ← RF(3)−RF(2)

FUNCTION

Evaluates R' + Cn.

DESCRIPTION

Data on the R bus is inverted and added with carry. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
No	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Increments if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Convert the data on the DA bus to two's complement and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0111	XX XXXX	XX XXXX	1 XX	00 0100	0	0000	10	X	X	XXXX	0	1	110

Assume register file 1 holds 3791FEF6 (Hex):

Source 0011 0111 1001 0001 1111 1110 1111 0110 R ← DA

Destination 1100 1000 0110 1110 0000 0001 0000 1010 RF(4) ← R' + Cn

FUNCTION

Evaluates S' + Cn.

DESCRIPTION

Data on the S bus is inverted and added to the carry. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Increments if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Convert the data on the MQ register to one's complement and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0101	XX XXXX	XX XXXX	X 11	00 0100	0	0000	10	X	X	XXXX	0	0	110

Assume MQ register file 1 holds 3791FEF6 (Hex):

Source: 0011 0111 1001 0001 1111 1110 1111 0110 — S ← MQ register

Destination: 1100 1000 0110 1110 0000 0001 0000 1001 — RF(4) ← S' + Cn

FUNCTION

Increments R if the carry is set.

DESCRIPTION

Data on the R bus is added to the carry. The sum appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands (MSH)

RF (B5-B0)	DB-Port	MQ Register
No	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Increments R if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
Cn	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Increment the data on the DA bus and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0110	XX XXXX	XX XXXX	1 XX	00 0100	0	0000	10	X	X	XXXX	0	1	110

Assume register file 1 holds 3791FEF6 (Hex).

Source 0001 0111 1001 0001 1111 1110 1111 0110 R ← DA

Destination 0001 0111 1001 0001 1111 1110 1111 0111 RF(4) ← R + Cn

FUNCTION

Increments S if the carry is set.

DESCRIPTION

Data on the S bus is added to the carry. The sum appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Increments S if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Increment the data in the MQ register and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0100	XX XXXX	XX XXXX	X 11	00 0100	0	0000	10	X	X	XXXX	0	1	110

Assume MQ register holds 54FF00FF (Hex):

Source 0101 0100 1111 1111 0000 0000 1111 1111 S ← MQ register

Destination 0101 0100 1111 1111 0000 0001 0000 0000 RF(4) ← S + Cn

FUNCTION

Load divide/BCD flip-flops from external data input.

DESCRIPTION

Uses an internal bypass path to load data from the S MUX directly into the divide/BCD flip-flops.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
No	No	No	No	No

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals

ZERO	= 0
N	= 0
OVR	= 0
C	= 0

EXAMPLE (assumes a 32-bit configuration)

Load the divide/BCD flip-flops with data from the DB input bus.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0000 1111	XX XXXX	XX XXXX	X 10	XX XXXX	X	XXXX	XX	X	X	XXXX	X	X	110

Assume DB input holds 2A08C618 (Hex):

Source 0010 1010 0000 1000 1100 0110 0001 1000 S ← DB bus

Destination 0010 1010 0000 1000 1100 0110 0001 1000 Divide/BCD flip-flops ← S

FUNCTION

Passes the result of the ALU instruction specified in the lower nibble of the instruction field to Y and the MQ register.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y and the MQ register.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
None	None

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Outputs MQ0 (LSB)
$\overline{SIO0}'$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB of result = 1
= 0 if MSB of result = 0
OVR = 1 if signed arithmetic overflow
C = 1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

Pass (Y ← F) and Load MQ with F — LOADMQ

EXAMPLE (assumes a 32-bit configuration)

Load the MQ register with data from register 1, and pass the data to the Y port.

(In this example, data is passed to the ALU by and INCR instruction without carry-in.)

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0110	00 0001	XX XXXX	0 XX	XX XXXX	0	XXXX	XX	X	X	XXXX	0	0	110

Assume register file 1 holds 2A08C618 (Hex):

Source: 0010 1010 0000 1000 1100 0110 0001 1000 — R ← RF(1)

Destination: 0010 1010 0000 1000 1100 0110 0001 1000 — MQ register ← R + Cn

FUNCTION

Passes the result of the ALU instruction specified in the upper nibble of the instruction field to Y MUX. Performs a circular left shift on MQ.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y MUX.

The contents of the MQ register are rotated one bit to the left. The MSB is rotated out and passed to the LSB of the same word, which may be 1, 2, or 4 bytes long.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the MQ register. If SSF is low, the MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
None	Circular Left

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high or floating; retains MQ without shift if low.
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Affects arithmetic operation programmed in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Add data in register 1 to data on the DB bus with carry-in and store the unshifted result in register 1. Circular shift the contents of the MQ register one bit to the left.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1101 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	1	110

Assume register file 1 holds 2508C618 (Hex), DB bus holds 11007530 (Hex), and MQ register holds 4DA99A0E (Hex).

Source	0010 0101 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0001 0001 0000 0000 0111 0101 0011 0000	S ← DB bus
Destination	0011 0110 0000 1001 0011 1011 0100 1001	RF(1) ← R + S + Cn
Source	0100 1101 1010 1001 1001 1010 0000 1110	MQ shifter ← MQ register
Destination	1001 1011 0101 0011 0011 0100 0001 1100	MQ register ← MQ shifter

FUNCTION

Passes the result of the ALU instruction specified in the upper nibble of the instruction field to Y MUX. Performs a left shift on MQ.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y MUX.

The contents of the MQ register are shifted one bit to the left. A zero is filled into the least significant bit of each word unless the $\overline{SIO}$ input for that word is programmed low; this will force the least significant bit to one. The MSB is dropped from each word, which may be 1, 2, or 4 bytes long, depending on the configuration selected.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the MQ register. If SSF is low, the MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
None	Logical Left

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high or floating; retains MQ without shift if low.
$\overline{SIO0}$	Yes	Fills a zero in LSB of MQ shifter if high or floating; sets LSB to one if low.
$\overline{SIO1}$	No	Inactive in 32-bit configuration; used in
$\overline{SIO2}$	No	configurations to select end-fill in LSBs.
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation programmed in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Add data in register 7 to data on the DB bus with carry-in and store the unshifted result in register 7. Shift the contents of the MQ register one bit to the left, filling a zero into the least significant bit.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	OES	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
1100 0001	00 0111	XX XXXX	0 10	00 0111	0	0000	10	X	X	XXXX	0	1	110	1111	0000

Assume register file 7 holds 7308C618 (Hex), DB bus holds 54007530 (Hex), and MQ register holds 61A99A0E (Hex).

Source	0111 0011 0000 1000 1100 0110 0001 1000	R ← RF(7)
Source	0101 0100 0000 0000 0111 0101 0011 0000	S ← DB bus
Destination	1100 0111 0000 1001 0011 1011 0100 1001	RF(7) ← R + S + Cn
Source	0110 0001 1010 1001 1001 1010 0000 1100	MQ shifter ← MQ register
Destination	1100 0011 0101 0011 0011 0100 0001 1000	MQ register ← MQ shifter

FUNCTION

Passes the result of the ALU instruction specified in the upper nibble of the instruction field to Y MUX. Performs an arithmetic right shift on MQ.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y MUX.

The contents of the MQ register are rotated one bit to the right. The sign bit of the most significant byte is retained. Bit 0 of the least significant byte is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the MQ register. If SSF is low, the MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
None	Arithmetic Right

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high or floating; retains MQ without shift if low.
$\overline{SIO0}$	No	Outputs LSB of MQ shifter (inverted).
$\overline{SIO1}$	No	Inactive in 32-bit configurations; used in other configurations to output LSBs from MQ shifter (inverted).
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation programmed in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Add data in register 1 to data in register 10 with carry-in and store the unshifted result in register 1. Shift the contents of the MQ register one bit to the right, retaining the sign bit.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1010 0001	00 0001	00 1010	0 00	00 0001	0	0000	10	X	X	XXXX	0	1	110

Assume register file 1 holds 5608C618 (Hex), register file 10 holds 14007530 (Hex), and MQ register holds 98A99A0E (Hex).

Source	0101 0110 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0001 0100 0000 0000 0111 0101 0011 0000	S ← RF(10)
Destination	0110 1010 0000 1001 0011 1011 0100 1001	RF(1) ← R + S + Cn
Source	1001 1000 1010 1001 1001 1010 0000 1110	MQ shifter ← MQ register
Destination	1100 1100 0101 0100 1100 1101 0000 0111	MQ register ← MQ shifter

FUNCTION

Passes the result of the ALU instruction specified in the upper nibble of the instruction field to Y MUX. Performs a right shift on MQ.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y MUX.

The contents of the MQ register are shifted one bit to the right. A zero is placed in the sign bit of the most significant byte unless the $\overline{SIO}$ input for that byte is set to zero; this will force the sign bit to 1. Bit 0 of the least significant byte is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the MQ register. If SSF is low, the MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
None	Logical Right

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high or floating; retains MQ without shift if low.
$\overline{SIO0}$	Yes	Fills a zero in LSB of MQ shifter if high or floating; sets LSB to one if low.
$\overline{SIO1}$	No	Inactive in 32-bit configuration; used in other configurations to select end-fill in LSBs.
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation programmed in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out = 1

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Add data in register 1 to data on the DB bus with carry-in and store the unshifted result in register 1. Shift the contents of the MQ register one bit to the left.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- EA EB0	Dest Addr C5-C0	Destination Selects SELMQ	WE3- WE0	SELRF1- SELRF0	OEA	OEB	OEY3- OEY0	OES	Cn	CF2- CF0
1011 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	1	110

Assume register file 1 holds 5608C618 (Hex), DB bus holds 14007530 (Hex), and MQ register holds 98A99A0E (Hex).

Source	0101 0110 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0001 0100 0000 0000 0111 0101 0011 0000	S ← DB bus
Destination	0110 1010 0000 1001 0011 1011 0100 1001	RF(1) ← R + S + Cn
Source	1001 1000 1010 1001 1001 1010 0000 1110	MQ shifter ← MQ register
Destination	0100 1100 0101 0100 1100 1101 0000 0111	MQ register ← MQ shifter

FUNCTION

Evaluates the logical expression R NAND S.

DESCRIPTION

Data on the R bus is NANDed with data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn		Inactive

Status Signals†

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 0
C	= 0

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Logically NAND the contents of register 3 and register 5, and store the result in register 5.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1100	00 0011	00 0101	0 00	00 0101	0	0000	10	X	X	XXXX	0	X	110

Assume register file 1 holds 60F6D840 (Hex) and register file 5 holds 13F6D377 (Hex).

Source 0110 0000 1111 0110 1101 1000 0100 0000 R ← RF(3)

Source 0001 0011 1111 0110 1101 0011 0111 0111 S ← RF(5)

Destination 1111 1111 0000 1001 0010 1111 1011 1111 RF(5) ← R NAND S

FUNCTION

Forces ALU output to zero.

DESCRIPTION

This instruction forces the ALU output to zero. The BCD flip-flops retain their old value. Note that the clear instruction (CLR) forces the ALU output to zero and clears the BCD flip-flops.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
No	No	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Status Signals

ZERO = 1
N = 0
OVR = 0
C = 0

EXAMPLE (assumes a 32-bit configuration)

Clear register 12.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1111	XX XXXX	XX XXXX	X XX	00 1100	0	0000	10	X	X	XXXX	0	X	110

Destination | 0000 0000 0000 0000 0000 0000 0000 0000 | RF(12) ← 0

FUNCTION

Evaluates the logical expression R NOR S.

DESCRIPTION

Data on the R bus is NORed with data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Inactive

Status Signals†

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 0
C	= 0

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Logically NOR the contents of register 3 and register 5, and store the result in register 5.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1011	00 0011	00 0101	0 00	00 0101	0	0000	10	X	X	XXXX	0	X	110

Assume register file 3 holds 60F6D840 (Hex) and register file 5 holds 13F6D377 (Hex).

Source 0110 0000 1111 0110 1101 1000 0100 0000 R ← RF(3)

Source 0001 0011 1111 0110 1101 0011 0111 0111 S ← RF(5)

Destination 1000 1100 0000 1001 0010 0100 1000 1000 RF(5) ← R NOR S

FUNCTION

Evaluates the logical expression R OR S.

DESCRIPTION

Data on the R bus is ORed with data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Inactive

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 0
C = 0

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Logically OR the contents of register 5 and register 3, and store the result in register 3.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1011	00 0101	00 0011	0 00	00 0011	0	0000	10	X	X	XXXX	0	X	110

Assume register file 5 holds 60F6D840 (Hex) and register file 3 holds 13F6D377 (Hex).

Source 0110 0000 1111 0110 1101 1000 0100 0000 R ← RF(5)

Source 0001 0011 1111 0110 1101 0011 0111 0111 S ← RF(3)

Destination 0111 0011 1111 0110 1101 1011 0111 0111 RF(3) ← R OR S

FUNCTION

Passes the result of the ALU instruction specified in the lower nibble of the instruction field to Y MUX.

DESCRIPTION

The result of the arithmetic or logical operation specified in the lower nibble of the instruction field (I3-I0) is passed unshifted to Y MUX.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB of result = 1
= 0 if MSB of result = 0
OVR = 1 if signed arithmetic overflow
C = 1 if carry-out condition

†C is ALU carry out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Add data in register 1 to data on the DB bus with carry-in and store the unshifted result in register 10.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0001	00 0001	XX XXXX	0 10	00 1010	0	0000	10	X	X	XXXX	0	1	110

Assume register file 3 holds 9308C618 (Hex) and DB bus holds 24007530 (Hex).

Source 1001 0011 0000 1000 1100 0110 0001 1000 R ← RF(1)

Source 0010 0100 0000 0000 0111 0101 0011 0000 S ← DB bus

Destination 1011 0111 0000 1001 0011 1011 0100 1001 RF(10) ← R + S + Cn

FUNCTION

Performs one of N-2 iterations of nonrestoring signed division by a test subtraction of the N-bit divisor from the 2N-bit dividend. An algorithm using this instruction is given in the "Other Arithmetic Instructions" section.

DESCRIPTION

SDIVI performs a test subtraction of the divisor from the dividend to generate a quotient bit. The test subtraction passes if the remainder is positive and fails if negative. If it fails, the remainder will be corrected during the next instruction.

SDIVI checks the pass/fail result of the test subtraction from the previous instruction, and evaluates

$F \leftarrow R + S$ if the test fails
$F \leftarrow R' + S + Cn$ if the test passes

A double precision left shift is performed; bit 7 of the most significant byte of the MQ shifter is transferred to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte of the ALU shifter is lost. The unfixed quotient bit is circulated into the least significant bit of the MQ shifter.

The R bus must be loaded with the divisor, the S bus with the most significant half of the result of the previous instruction (SDIVI during iteration or SDIVIS at the beginning of iteration). The least significant half of the previous result is in the MQ register. Carry-in should be programmed high. Overflow occurring during SDIVI is reported to OVR at the end of the signed divide routine (after SDIVQF).

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Pass internally generated end-fill bits.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed high

Status Signals

ZERO = 1 if intermediate result = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Initializes 'ACT8832 for nonrestoring signed division by shifting the dividend left and internally preserving the sign bit. An algorithm using this instruction is given in the "Other Arithmetic Instructions section.

DESCRIPTION

This instruction prepares for signed divide iteration operations by shifting the dividend and storing the sign for future use.

The preceding instruction should load the MQ register with the least significant half of the dividend. During SDIVIN, the S bus should be loaded with the most significant half of the dividend, and the R bus with the divisor. Y-output should be written back to the register file for use in the next instruction.

A double precision logical left shift is performed; bit 7 of the most significant byte of the MQ shifter is transferred to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte of the ALU shifter is lost. The unfixed quotient sign bit is shifted into the least significant bit of the MQ shifter.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Pass internally generated end-fill bits.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Inactive

Status Signals

ZERO = 1 if divisor = 0
N = 0
OVR = 0
Cn = 0

FUNCTION

Computes the first quotient bit of nonrestoring signed division. An algorithm using this instruction is given in the "Other Arithmetic Instructions" section..

DESCRIPTION

SDIVIS computes the first quotient bit during nonrestoring signed division by subtracting the divisor from the dividend, which was left-shifted during the prior SDIVIN instruction. The resulting remainder due to subtraction may be negative. If so, the subsequent SDIVI instruction will restore the remainder during the next subtraction.

The R bus must be loaded with the divisor and the S bus with the most significant half of the remainder. The result on the Y bus should be loaded back into the register file for use in the next instruction. The least significant half of the remainder is in the MQ register. Carry-in should be programmed high.

A double precision left shift is performed; bit 7 of the most significant byte of the MQ shifter is transferred to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte of the ALU shifter is lost. The unfixed quotient bit is circulated into the least significant bit of the MQ shifter.

Overflow occurring during SDIVIS is reported to OVR at the end of the signed division routine (after SDIVQF).

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Pass internally generated end-fill bits.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed high.

Status Signals

ZERO = 1 if intermediate result = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Solves the final quotient bit during nonrestoring signed division. An algorithm using this instruction is given in the "Other Arithmetic Instructions" section.

DESCRIPTION

SDIVIT performs the final subtraction of the divisor from the remainder during nonrestoring signed division. SDIVIT is preceded by N-2 iterations of SDIVI, where N is the number of bits in the dividend.

The R bus must be loaded with the divisor, and the S bus must be loaded with the most significant half of the result of the last SDIVI instruction. The least significant half lies in the MQ register. The Y bus result must be loaded back into the register file for use in the subsequent DIVRF instruction. Carry-in should be programmed high.

SDIVIT checks the pass/fail result of the previous instruction's test subtraction and evaluates;

$Y \leftarrow R + S$ if the test fails
$Y \leftarrow R' + S + Cn$ if the test passes

The contents of the MQ register are shifted one bit to the left; the unfixed quotient bit is circulated into the least significant bit.

Overflow during this instruction is reported to OVR at the end of the signed division routine (after SDIVQF).

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Pass internally generated end-fill bits.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed high

Status Signals

ZERO = 1 if intermediate result = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Tests for overflow during nonrestoring signed division. An algorithm using this instruction is given in the "Other Arithmetic Instructions section.

DESCRIPTION

This instruction performs an initial test subtraction of the divisor from the dividend. If overflow is detected, it is preserved internally and reported at the end of the divide routine (after SDIVQF). If overflow status is ignored, the SDIVO instruction may be omitted.

The divisor must be loaded onto the R bus; the most significant half of the previous SDIVIN result must be loaded onto the S bus. The least significant half is in the MQ register.

The result on the Y bus should not be stored back into the register file; WE' should be programmed high.

Carry-in should also be programmed high.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Should be programmed high

Status Signals

ZERO = 1 if divisor = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Tests the quotient result after nonrestoring signed division and corrects it if necessary. An algorithm using this instruction is given in the "Other Arithmetic Instructions" section.

DESCRIPTION

SDIVQF is the final instruction required to compute the quotient of a 2N-bit dividend by an N-bit divisor. It corrects the quotient if the signs of the divisor and dividend are different and the remainder is nonzero.

The fix is implemented by incrementing S:

$Y \leftarrow S + 1$ if a fix is required
$Y \leftarrow S + 0$ if no fix is required

The R bus must be loaded with the divisor, and the S bus with the most significant half of the result of the preceding DIVRF instruction. The least significant half is in the MQ register.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Should be programmed high

Status Signals

ZERO = 1 if quotient = 0
N = 1 if sign of quotient + 1
= 0 if sign of quotient + 0
OVR = 1 if divide overflow
C = 1 if carry-out

FUNCTION

Selects S if SSF is high; otherwise selects R.

DESCRIPTION

Data on the S bus is passed to Y if SSF is programmed high or floating; data on the R bus is passed without carry to Y if SSF is programmed low.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands (MSH)

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Selects S if high, R if low.
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	No	Inactive

Status Signals

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 0
C	= 0

EXAMPLE (assumes a 32-bit configuration)

Compare the two's complement numbers in registers 1 and 3 and store the larger in register 5.

1. Subtract (SUBS) data in register 3 from data in register 1 and pass the result to the Y bus.
2. Perform Select S/R instruction and pass result to register 5.

[This example assumes the SSF is set by the negative status (N) from the previous instruction].

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 0011	00 0001	00 0011	0 00	XX XXXX	0	XXXX	XX	X	X	0000	0	1	110
0001 0000	00 0001	00 0011	0 00	00 0101	0	0000	10	X	X	XXXX	0	0	110

Assume register file 1 holds 008497D0 (Hex) and register file 3 holds 01C35250 (Hex).

Instruction Cycle 1

Source `0000 0000 1000 0100 1001 0111 1101 0000` R ← RF(1)

Source `0000 0001 1100 0011 0101 0010 0101 0000` S ← RF(3)

Destination `1111 1110 1100 0001 0100 0101 1000 0000` Y bus ← R + S' + Cn

`1` N ← 1

Instruction Cycle 2

Source `0000 0000 1000 0100 1001 0111 1101 0000` R ← RF(1)

`1` SSF ← 1

Source `0000 0001 1100 0011 0101 0010 0101 0000` S ← RF(3)

Destination `0000 0001 1100 0011 0101 0010 0101 0000` RF(5) ← S

FUNCTION

Resets bits in selected bytes of S-bus data using mask in C3-C0::A3-A0.

DESCRIPTION

The register addressed by B5-B0 is both the source and destination for this instruction. The source word is passed on the S bus to the ALU, where it is compared to an 8-bit mask, consisting of a concatenation of the C3-C0 and A3-A0 address ports (C3-C0::A3-A0). The mask is input via the R bus. All bits in the source word that are in the same bit position as ones in the mask are reset. Bytes with their $\overline{SIO}$ inputs programmed low perform the Reset Bit instruction. Bytes with their $\overline{SIO}$ inputs programmed high or floating pass S unaltered.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	Yes

Available S Bus Source Operands (MSH)

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
No	Yes	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Byte-select
$\overline{SIO1}$	No	Byte-select
$\overline{SIO2}$	No	Byte-select
$\overline{SIO3}$	No	Byte-select
Cn	No	Inactive

Status Signals

ZERO	= 1 if result (selected bytes) = 0
N	= 0
OVR	= 0
C	= 0

EXAMPLE (assumes a 32-bit configuration)

Set bits 3-0 of bytes 1 and 2 of register file 8 to zero and store the result back in register 8.

Instr Code I7-I0	Mask (LSH) A3-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Mask (MSH) C3-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0001 1000	1111	00 1000	X 00	0000	0	0000	10	X	X	XXXX	0	X	110	1001	0000

Assume register file 8 holds A083BEBE (Hex).

Source	0000 1111 0000 1111 0000 1111 0000 1111	Rn ← C3-C0::A3-A0
Source	1010 0000 1000 0011 1011 1110 1011 1110	Sn ← RF(3)n
ALU	1010 0000 1000 0000 1011 0000 1011 1110	Fn ← Sn AND Rn
Destination	1010 0000 1000 0000 1011 0000 1011 1110	RF(8)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 8 gets F if byte selected, S if byte not selected.

FUNCTION

Sets bits in selected bytes of S-bus data using mask in C3-C0::A3-A0.

DESCRIPTION

The register addressed by B5-B0 is both the source and destination for this instruction. The source word is passed on the S bus to the ALU, where it is compared to an 8-bit mask, consisting of a concatenation of the C3-C0 and A3-A0 address ports (C3-C0::A3-A0). The mask is input via the R bus. All bits in the source word that are in the same bit position as ones in the mask are forced to a logical one. Bytes with their $\overline{SIO}$ inputs programmed low perform the Set Bit instruction. Bytes with their $\overline{SIO}$ inputs programmed high or floating pass S unaltered.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	Yes

Available S Bus Source Operands (MSH)

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
No	Yes	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte-select
$\overline{SIO1}$	No	Byte-select
$\overline{SIO2}$	No	Byte-select
$\overline{SIO3}$	No	Byte-select
Cn	No	Inactive

Status Signals

ZERO = 1 if result (selected bytes) = 0
N = 0
OVR = 0
C = 0

EXAMPLE (assumes a 32-bit configuration)

Set bits 3-0 of byte 1 of register file 1 to zero and store the result back in register 1.

Instr Code I7-I0	Mask (LSH) A3-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Mask (MSH) C3-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0000 1000	1111	00 0001	X 00	0000	0	0000	10	X	X	XXXX	0	X	110	1101	0000

Assume register file 8 holds A083BEBE (Hex).

Source	0000 1111 0000 1111 0000 1111 0000 1111	Rn ← C3-C0::A3-A0
Source	1010 0000 1000 0011 1011 1110 1011 1110	Sn ← RF(1)n
ALU	1010 0000 1000 0011 1011 1111 1011 1110	Fn ← Sn OR Rn
Destination	1010 0000 1000 0011 1011 1111 1011 1110	RF(1)n ← Fn or Sn†

†F = ALU result
n = nth byte
Register file 1 gets F if byte selected, S if byte not selected.

FUNCTION

Performs arithmetic left shift on result of ALU operation specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is shifted one bit to the left. A zero is filled into bit 0 of the least significant byte of each word unless the $\overline{SIO}$ input is programmed low; this will force bit 0 to one. Bit 7 is dropped from the most significant byte in each word, which may be 1, 2, or 4 bytes long, depending on the configuration selected.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the MQ register. If SSF is low, the MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Arithmetic Left	None

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result if low.
$\overline{SIO0}$	Yes	Fills a zero in LSB of each word if high; fills a
$\overline{SIO1}$	Yes	one in LSB if low.
$\overline{SIO2}$	Yes	
$\overline{SIO3}$	Yes	
Cn	No	Affects arithmetic operation programmed in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow or if MSB XOR MSB-1 = 1 before shift
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform the computation A = 2(A + B), where A and B are single-precision, two's complement numbers. Let A be stored in register 1 and B be input via the DB bus.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	OEB	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$	SSF
0100 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	0	110	1110	0000	1

Assume register file 1 holds 1308C618 (Hex), DB bus holds 44007530 (Hex).

Source	0001 0011 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0100 0100 0000 0000 0111 0101 0011 0000	S ← DB bus
Intermediate Result	0101 0111 0000 1001 0011 1011 0100 1000	ALU Shifter ← R + S + Cn
Destination	1010 1110 0001 0010 0111 0110 1001 0001	RF(1) ← ALU shift result

FUNCTION

Performs arithmetic left shift on MQ register (LSH) and result of ALU operation (MSH) specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is used as the upper half of a double-precision word, the contents of the MQ register as the lower half.

The contents of the MQ register are shifted one bit to the left. A zero is filled into bit 0 of the least significant byte of each word unless the $\overline{SIO}$ input for the word is set to zero; this will force bit 0 to one. Bit 7 of the most significant byte in the MQ shifter is passed to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte in the ALU shifter is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX and MQ register. If SSF is low, the ALU output and MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Arithmetic Left	Arithmetic Left

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result if low.
$\overline{SIO0}$	Yes	Fills a zero in LSB of each word if high; fills a
$\overline{SIO1}$	Yes	one in LSB if low.
$\overline{SIO2}$	Yes	
$\overline{SIO3}$	Yes	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow or if MSB XOR MSB-1 = 1 before shift
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform the computation A = 2(A + B), where A and B are two's complement numbers. Let A be a double precision number residing in register 1 (MSH) and the MQ register (LSH). Let B be a single precision number which is input through the DB bus.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- EA EB0	Dest Addr C5-C0	Destination Selects SELMQ	WE3- WE0	SELRF1- SELRF0	OEA	OEB	OEY3- OEY0	OES	Cn	CF2- CF0	SIO3- SIO0	IESIO3- IESIO0	SSF
0101 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	0	110	1110	0000	1

Assume register file 1 holds 2408C618 (Hex), DB bus holds 26007530 (Hex), and MQ register holds 50A99A0E (Hex).

MSH

Source	0010 0100 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0010 0110 0000 0000 0111 0101 0011 0000	S ← DB bus
Intermediate Result	0100 1010 0000 1001 0011 1011 0100 1000	ALU Shifter ← R + S + Cn
Destination	1001 0100 0001 0010 0111 0110 1001 0000	RF(1) ← ALU shift register

LSH

Source	0101 0000 1010 1001 1001 1010 0000 1110	MQ shifter ← MQ register
Destination	1010 0001 0101 0011 0011 0100 0001 1101	MQ register ← MQ shift result

FUNCTION

Performs circular left shift on result of ALU operation specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is rotated one bit to the left. Bit 7 of the most significant byte in each word is passed to bit 0 of the least significant byte in the word, which may be 1, 2, or 4 bytes long.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to Y MUX. If SSF is low, F is passed unaltered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Circular Left	None

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result if low.
$\overline{SIO0}$	No	Bit 7 of ALU result
$\overline{SIO1}$	No	Bit 15 of ALU result
$\overline{SIO2}$	No	Bit 23 of ALU result
$\overline{SIO3}$	No	Bit 31 of ALU result
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a circular left shift of register 6 and store the result in register 1.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	SSF
0110 0110	00 0110	XX XXXX	0 00	00 0001	0	0000	10	X	X	XXXX	0	0	110	1

Assume register file 6 holds 3788C618 (Hex).

Source: 0011 0111 1000 1000 1100 0110 0001 1000 — R ← RF(6)

Intermediate Result: 0011 0111 1000 1000 1100 0110 0001 1000 — ALU Shifter ← R + Cn

Destination: 0110 1111 0001 0001 1000 1100 0011 0000 — RF(1) ← ALU shifter result

FUNCTION

Performs circular left shift on MQ register (LSH) and result of ALU operation specified in lower nibble of instruction field (MSH).

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is used as the upper half of a double-precision word, the contents of the MQ register as the lower half.

The contents of the MQ and ALU registers are rotated one bit to the left. Bit 7 of the most significant byte in the MQ shifter is passed to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte is passed to bit 0 of the least significant byte in the MQ shifter.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to Y MUX. If SSF is low, F is passed unaltered and the MQ register is not changed.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Circular Left	Circular Left

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result if low.
$\overline{SIO0}$	No	Bit 7 of ALU result
$\overline{SIO1}$	No	Bit 15 of ALU result
$\overline{SIO2}$	No	Bit 23 of ALU result
$\overline{SIO3}$	No	Bit 31 of ALU result
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a circular left double precision shift of data in register 6 (MSH) and MQ (LSH), and store the result back in register 6 and the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	SSF
0111 0110	00 0110	XX XXXX	0 00	00 0110	0	0000	10	X	X	XXXX	0	0	110	1

Assume register file 6 holds 3708C618 (Hex) and MQ register holds 50A99A0E (Hex).

MSH

Source	0011 0111 0000 1000 1100 0110 0001 1000	R ← RF(6)
Intermediate Result	0011 0111 0000 1000 1100 0110 0001 1000	ALU Shifter ← R + Cn
Destination	0110 1111 0001 0001 1000 1100 0011 0000	RF(6) ← ALU shifter result

LSH

Source	0101 0000 1010 1001 1001 1010 0000 1110	MQ register ← MQ register
Destination	1010 0001 0101 0011 0011 0100 0001 1100	MQ register ← MQ shift result

FUNCTION

Converts data on the S bus from sign magnitude to two's complement or vice versa.

DESCRIPTION

The S bus provides the source word for this instruction. The number is converted by inverting S and adding the result to the carry-in, which should be programmed high for proper conversion; the sign bit of the result is then inverted. An error condition will occur if the source word is a negative zero (negative sign and zero magnitude). In this case, SMTC generates a positive zero, and the OVR pin is set high to reflect an illegal conversion.

The sign bit of the selected operand in the most significant byte is tested; if it is high, the converted number is passed to the destination. Otherwise the operand is passed unaltered.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Should be programmed high for proper conversion

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 1 if input of most significant byte is 80 (Hex) and results in all other bytes are 00 (Hex).
C = 1 if S = 0

EXAMPLES (assumes a 32-bit configuration)

Convert the two's complement number in register 1 to sign magnitude representation and store the result in register 4.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0101 1000	XX XXXX	00 0001	X 00	00 0100	0	0000	10	X	X	XXXX	0	1	110

Example 1: Assume register file 1 holds C3F6D840 (Hex).

Source: 1100 0011 1111 0110 1101 1000 0100 0000 S ← RF(1)

Destination: 1011 1100 0000 1001 0010 0111 1100 0000 RF(4) ← S' + Cn

Example 2: Assume register file 1 holds 550927C0 (Hex).

Source: 0101 0101 0000 1001 0010 0111 1100 0000 S ← RF(1)

Destination: 0101 0101 0000 1001 0010 0111 1100 0000 RF(4) ← S

FUNCTION

Computes one of N-1 signed or N mixed multiplication iterations for computing an N-bit by N-bit product. Algorithms for signed and mixed multiplication using this instruction are given in the "Other Arithmetic Instructions" section.

DESCRIPTION

SMULI checks to determine whether the multiplicand should be added with the present partial product. The instruction evaluates:

$F \leftarrow R + S + Cn$	if the addition is required
$F \leftarrow S$	if no addition is required

A double precision right shift is performed. Bit 0 of the least significant byte of the ALU shifter is passed to bit 7 of the most significant byte of the MQ shifter; carry-out is passed to the most significant bit of the ALU shifter.

The S bus should be loaded with the contents of an accumulator and the R bus with the multiplicand. The Y bus result should be written back to the accumulator after each iteration of UMULI. The accumulator should be cleared and the MQ register loaded with the multiplier before the first iteration.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
Right	Right

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes LSB from ALU shifter to MSB of MQ shifter.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed low

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 0
C = 1 if carry-out

FUNCTION

Performs the final iteration for computing an N-bit by N-bit signed product. An algorithm for signed multiplication using this instruction is given in the "other Arithmetic Instructions" section.

DESCRIPTION

SMULI checks the present multiplier bit (the least significant bit of the MQ register) to determine whether the multiplicand should be added with the present partial product. The instruction evaluates:

$F \leftarrow R' + S + Cn$	if the addition is required
$F \leftarrow S$	if no addition is required

with the correct sign in the product.

A double precision right shift is performed. Bit 0 of the least significant byte of the ALU shifter is passed to bit 7 of the most significant byte of the MQ shifter.

The S bus should be loaded with the contents of an register file holding the previous iteration result; the R bus must be loaded with the multiplicand. After executing SMULT, the Y bus contains the most significant half of the product, and MQ contains the least significant half.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	No

Shift Operations

ALU	MQ
Right	Right

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes LSB from ALU shifter to MSB of MQ shifter.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed low

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 0
C = 1 if carry-out

FUNCTION

Tests the two most significant bits of the MQ register. If they are the same, shifts the number to the left.

DESCRIPTION

This instruction is used to normalize a two's complement number in the MQ register by shifting the number one bit position to the left and filling a zero into the LSB (unless the $\overline{SIO}$ input for that word is low). Data on the S bus is added to the carry, permitting the number of shifts performed to be counted and stored in one of the register files.

The shift and the S bus increment are inhibited whenever normalization is attempted on a number already normalized. Normalization is complete when overflow occurs.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	No

Available S Bus Source Operands (Count)

RF (B5-B0)	DB-Port	MQ Register
Yes	No	No

Available Destination Operands (Count)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations (Conditional)

ALU	MQ
No	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes internally generated end-fill bit.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Increments S bus (shift count) if set to one.

Status Signals

ZERO = 1 if result = 0
N = 1 if MSB of MQ register = 1
OVR = 1 if MSB of MQ register XOR 2nd MSB = 1
C = 1 if carry-out = 1

EXAMPLE (assumes a 32-bit configuration)

Normalize the number in the MQ register, storing the number of shifts in register 3.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0010 0000	XX XXXX	00 0011	X 00	00 0011	0	0000	10	X	X	XXXX	0	1	110

Assume register file 3 holds 00000003 (Hex) and MQ register holds 3699D84E (Hex).

Operand

Source: 0011 0110 1001 1001 1101 1000 0100 1110 — MQ shifter ← MQ register

Destination: 0110 1101 0011 0011 1011 0000 1001 1100 — MQ register ← MQ shifter

Count

Source: 0000 0000 0000 0000 0000 0000 0000 0011 — S ← RF(3)

Destination: 0000 0000 0000 0000 0000 0000 0000 0100 — RF(3) ← S + Cn

FUNCTION

Performs arithmetic right shift on result of ALU operation specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is shifted one bit to the right. The sign bit of the most significant byte is retained unless it is inverted as a result of overflow. Bit 0 of the least significant byte is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX. If SSF is low, the ALU result will be passed unshifted to the Y MUX.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Arithmetic Right	None

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shifted output if high; passes ALU result if low.
$\overline{SIO0}$	No	LSB is shifted out from each word, which may be 1, 2, or 4 bytes long depending on selected configuration
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB of result = 1
= 0 if MSB of result = 0
OVR = 0
C = 1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform the computation A = (A + B)/2, where A and B are single-precision numbers. Let A reside in register 1 and B be input via the DB bus.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	SSF
0000 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	0	110	1

Assume register file 1 holds 6A08C618 (Hex) and DB bus holds 51007530 (Hex).

Source: 0110 1010 0000 1000 1100 0110 0001 1000 — R ← RF(1)

Source: 0101 0001 0000 0000 0111 0101 0011 0000 — S ← DB bus

Intermediate‡ Result: 1011 1011 0000 1001 0011 1011 0100 1000 — ALU Shifter ← R + S + Cn

Destination: 0101 1101 1000 0100 1001 1101 1010 0100 — RF(1) ← ALU shift result

‡After the intermediate operation (ADD), overflow has occurred and OVR status signal is set high. When the arithmetic right shift is executed, the sign bit is corrected (see Table 16 for shift definition notes).

FUNCTION

Performs arithmetic right shift on MQ register (LSH) and result of ALU operation (MSH) specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is used as the upper half of a double precision word, the contents of the MQ register as the lower half.

The contents of the ALU are shifted one bit to the right. The sign bit of the most significant byte is retained unless the sign bit is inverted as a result of overflow. Bit 0 of the least significant byte in the ALU shifter is passed to bit 7 of the most significant byte of the MQ register. Bit 0 of the MQ register's least significant byte is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX. If SSF is low, the ALU result will be passed unshifted to the Y MUX.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Arithmetic Right	Arithmetic Right

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shifted output if high; passes ALU result if low.
$\overline{SIO0}$	No	LSB of ALU shifter is passed to MSB of MQ shifter,
$\overline{SIO1}$	No	and LSB of MQ shifter is dropped.
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	0
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform the computation A = (A + B)/2, where A and B are two's complement numbers. Let A be a double precision number residing in register 1 (MSH) and MQ (LSH). Let B be a single precision number which is input through the DB bus.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	SSF
0001 0001	00 0001	XX XXXX	0 10	00 0001	0	0000	10	X	X	XXXX	0	0	110	1

Assume register file 1 holds 4A08C618 (Hex), and DB bus holds 51007530 (Hex), and MQ register holds 17299A0F (Hex).

MSH

Source	0100 1010 0000 1000 1100 0110 0001 1000	R ← RF(1)
Source	0101 0001 0000 0000 0111 0101 0011 0000	S ← DB bus
Intermediate‡ Result	1001 1011 0000 1001 0011 1011 0100 1000	ALU Shifter ← R + S + Cn
Destination	0100 1101 1000 0100 1001 1101 1010 0100	RF(1) ← ALU shift result

LSH

Source	0001 0111 0010 1001 1001 1010 0000 1111	MQ shifter ← MQ register
Destination	0000 1011 1001 0100 1100 1101 0000 0111	MQ register ← MQ shift result

‡After the intermediate operation (ADD), overflow has occurred and OVR status signal is set high. When the arithmetic right shift is executed, the sign bit is corrected (see Table 16 for shift definition notes).

FUNCTION

Performs circular right shift on result of ALU operation specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is shifted one bit to the right. Bit 0 of the least significant byte is passed to bit 7 of the most significant byte in the same word, which may be 1, 2, or 4 bytes long depending on the selected configuration.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX. If SSF is low, the ALU result will be passed unshifted to the Y MUX.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Circular Right	None

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result if low.
$\overline{SIO0}$	No	Rotates LSB to MSB of the same word, which may be 1, 2, or 4 bytes long depending on configuration
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a circular right shift of register 6 and store the result in register 1.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	SSF
1000 0110	00 0110	XX XXXX	0 XX	00 0001	0	0000	10	X	X	XXXX	0	0	110	1

Assume register file 6 holds 3788C618 (Hex).

Source	0011 0111 1000 1000 1100 0110 0001 1000	R ← RF(6)
Intermediate Result	0011 0111 1000 1000 1100 0110 0001 1000	ALU Shifter ← R + Cn
Destination	0001 1011 1100 0100 0110 0011 0000 1100	RF(1) ← ALU shift result

FUNCTION

Performs circular right shift on MQ register (LSH) and result of ALU operation (MSH) specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is used as the upper half of a double precision word, the contents of the MQ register as the lower half.

The contents of the ALU and MQ shifters are rotated one bit to the right. Bit 0 of the least significant byte in the ALU shifter is passed to bit 7 of the most significant byte of the MQ shifter. Bit 0 of the least significant byte is passed to bit 7 of the most significant byte of the ALU shifter.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MXU and MQ register. If SSF is low, the Y MUX and MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Circular Right	Circular Right

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result and retains MQ register if low.
$\overline{SIO0}$	No	Rotates LSB of ALU shifter to MSB of MQ shifter,
$\overline{SIO1}$	No	and LSB of MQ shifter to MSB of ALU shifter
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB of result = 1
	=	0 if MSB of result = 0
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a circular right double precision shift of the data in register 6 (MSH) and MQ (LSH), and store the result back in register 6 and the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1001 0110	00 0110	XX XXXX	0 XX	00 0110	0	0000	10	X	X	XXXX	0	0	110

Assume register file 6 holds 3788C618 (Hex) and MQ register holds 50A99A0F (Hex).

MSH

Source	0011 0111 0000 1000 1100 0110 0001 1000	R ← RF(6)
Intermediate Result	0011 0111 0000 1000 1100 0110 0001 1000	ALU shifter ← R + Cn
Destination	1001 1011 1000 0100 0110 0011 0000 1100	RF(6) ← ALU shift result

LSH

Source	0101 0000 1010 1001 1001 1010 0000 1111	MQ shifter ← MQ register
Destination	0010 1000 0101 0100 1100 1101 0000 0111	MQ register ← MQ shift result

FUNCTION

Performs logical right shift on result of ALU operation specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is shifted one bit to the right. A zero is placed in the bit 7 of the most significant byte of each word unless the $\overline{SIO}$ input for the word is programmed low; this will force the sign bit to one. The LSB is dropped from the word, which may be 1, 2, or 4 bytes long depending on selected configuration.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX. If SSF is low, the ALU result will be passed unshifted to the Y MUX.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Logical Right	None

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals‡

Signal	User Programmable	Use
SSF		Passes shift result if high or floating; passes ALU result if low.
$\overline{SIO0}$	Yes	Fills a zero in MSB of the word if high or floating;
$\overline{SIO1}$	Yes	fills a one in MSB if low.
$\overline{SIO2}$	Yes	
$\overline{SIO3}$	Yes	
Cn		Inactive

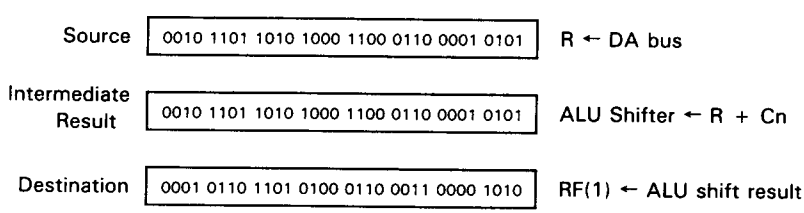
‡Cn is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a logical right single precision shift on data on the DA bus, and store the result in register 1.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$	SSF
0010 0110	XX XXXX	XX XXXX	1 XX	00 0001	0	0000	10	X	X	XXXX	0	0	110	XXX1	0000	1

Assume DA bus holds 2DA8C615.



FUNCTION

Performs logical right shift on MQ register (LSH) and result of ALU operation (MSH) specified in lower nibble of instruction field.

DESCRIPTION

The result of the ALU operation specified in instruction bits I3-I0 is used as the upper half of a double precision word, the contents of the MQ register as the lower half.

The ALU result is shifted one bit to the right. A zero is placed in the sign bit of the most significant byte unless the $\overline{SIO}$ input for that word is programmed low; this will force the sign bit to one. Bit 0 of the least significant byte is passed to bit 7 of the most significant byte of the MQ shifter. Bit 0 of the least significant byte of the MQ shifter is dropped.

The shift may be made conditional on SSF. If SSF is high or floating, the shift result will be sent to the Y MUX and MQ register. If SSF is low, the ALU result and MQ register will not be altered.

*A list of ALU operations that can be used with this instruction is given in Table 15.

Shift Operations

ALU Shifter	MQ Shifter
Logical Right	Logical Right

Available Destination Operands (ALU Shifter)

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	Yes	Passes shift result if high; passes ALU result and retains MQ
$\overline{SIO0}$	Yes	Fills a zero in MSB if high or floating;
$\overline{SIO1}$	Yes	fills a one MSB if low.
$\overline{SIO2}$	Yes	
$\overline{SIO3}$	Yes	
Cn	No	Affects arithmetic operation specified in bits I3-I0 of instruction field.

Status Signals†

ZERO	= 1 if result = 0
N	= 1 if MSB of result = 1
	= 0 if MSB of result = 0
OVR	= 1 if signed arithmetic overflow
C	= 1 if carry-out condition

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Perform a logical right double precision shift of the data in register 1 (MSH) and MQ (LSH), filling a one into the most significant bit, and store the result back in register 1 and the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0011 0110	XX XXXX	00 0001	X 00	00 0001	0	0000	10	X	X	XXXX	0	0	110	1110	0000

Assume register file 1 holds 2DA8C615 (Hex) and MQ register holds 50A99A0E (Hex).

MSH

Source	0010 1101 1010 1000 1100 0110 0001 0101	R ← RF(1)
Intermediate Result	0010 1101 1010 1000 1100 0110 0001 0101	ALU Shifter ← S + Cn
Destination	1001 0110 1101 0100 0110 0011 0000 1010	RF(1) ← ALU shift result

LSH

Source	0101 0000 1010 1001 1001 1010 0000 1110	MQ shifter ← MQ register
Destination	1010 1000 0101 0100 1100 1101 0000 0111	MQ register ← MQ shift result

FUNCTION

Subtracts four-bit immediate data on A3-A0 with carry from S-bus data.

DESCRIPTION

Immediate data in the range 0 to 15, supplied by the user at A3-A0, is inverted and added with carry to S.

Available R Bus Source Operands (Constant)

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	Yes	No	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	None

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Inactive
$\overline{SIO1}$	No	Inactive
$\overline{SIO2}$	No	Inactive
$\overline{SIO3}$	No	Inactive
Cn	Yes	Two's complement subtraction if programmed high.

Status Signals

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 1 if arithmetic signed overflow
C	= 1 if carry-out

EXAMPLE (assumes a 32-bit configuration)

Subtract the value 12 from data on the DB bus, and store the result into register file 1.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel $\overline{EA}$ EB1- EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
0111 1000	00 1100	XX XXXX	X 10	00 0001	0	0000	10	X	X	XXXX	0	1	110

Assume bits A3-A0 hold C (Hex) and DB bus holds 24000100 (Hex).

Source 0000 0000 0000 0000 0000 0000 0000 1100 R ← A3-A0

Source 0010 0100 0000 0000 0000 0001 0000 0000 S ← DB bus

Destination 0010 0100 0000 0000 0000 0000 1111 0100 RF(1) ← R′ + S + Cn

FUNCTION

Subtracts data on the R bus from S with carry.

DESCRIPTION

Data on the R bus is subtracted with carry from data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Two's complement subtraction if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Subtract data in register 1 from data on the DB bus, and store the result in the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1110 0010	00 0001	XX XXXX	0 10	XX XXXX	1	XXXX	XX	X	X	XXXX	0	1	110

Assume register file 1 holds 150084D0 (Hex) and DB bus holds 4900C350 (Hex).

Source 0001 0101 0000 0000 1000 0100 1101 0000 R ← RF(1)

Source 0100 1001 0000 0000 1100 0011 0101 0000 S ← DB bus

Destination 0011 0100 0000 0000 0011 1110 1000 0000 MQ register ← R' + S + Cn

FUNCTION

Subtracts data on the S bus from R with carry.

DESCRIPTION

Data on the S bus is subtracted with carry from data on the R bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Two's complement subtraction if programmed high.

Status Signals†

ZERO	=	1 if result = 0
N	=	1 if MSB = 1
OVR	=	1 if signed arithmetic overflow
C	=	1 if carry-out

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Subtract data on the DB bus from data in register 1, and store the result in the MQ register.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1110 0011	00 0001	XX XXXX	0 10	XX XXXX	1	XXXX	XX	X	X	XXXX	0	1	110

Assume register file 1 holds 150084D0 (Hex) and DB bus holds 4900C350 (Hex).

Source `0001 0101 0000 0000 1000 0100 1101 0000` R ← RF(1)

Source `0100 1001 0000 0000 1100 0011 0101 0000` S ← DB bus

Destination `1100 1011 1111 1111 1100 0001 1000 0000` MQ register ← R + S' + Cn

FUNCTION

Tests bits in selected bytes of S-bus data for zeros using mask in C3-C0::A3-A0.

DESCRIPTION

The S bus is the source word for this instruction. The source word is passed to the ALU, where it is compared to an 8-bit mask, consisting of a concatenation of the C3-C0 and A3-A0 address ports (C3-C0::A3-A0). The mask is input via the R bus. The test will pass if the selected byte has zeros at all bit locations specified by the ones of the mask. Bytes are selected by programming the $\overline{SIO}$ inputs low. Test results are indicated on the ZERO output, which goes to one if the test passes. Register write is internally disabled during this instruction.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	Yes

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte Select
$\overline{SIO1}$	Yes	Byte Select
$\overline{SIO2}$	Yes	Byte Select
$\overline{SIO3}$	Yes	Byte Select
Cn	No	Inactive

Status Signals

ZERO	= 1 if result (selected bytes) = Pass
N	= 0
OVR	= 0
C	= 0

EXAMPLE (assumes a 32-bit configuration)

Test bits 7, 6 and 5 of bytes 0 and 2 of data in register 3 for zeroes.

Instr Code I7-I0	Mask (LSH) A3-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Mask (MSH) C3-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0011 1000	0000	00 0011	X 00	1110	X	XXXX	XX	X	X	XXXX	0	X	110	1010	0000

Assume register file 3 holds 881CD003 (Hex).

Source | 1110 0000 1110 0000 1110 0000 1110 0000 | R ← Mask (C3-C0::A3-A0)

Source | 1000 1000 0001 1100 1101 0000 0000 0011 | SN ← RF(3)n†

Output | 1 | ZERO ← 1

†n = nth byte

FUNCTION

Tests bits in selected bytes of S-bus data for ones using mask in C3-C0::A3-A0.

DESCRIPTION

The S bus is the source word for this instruction. The source word is passed to the ALU, where it is compared to an 8-bit mask, consisting of a concatenation of the C3-C0 and A3-A0 address ports (C3-C0::A3-A0). The mask is input via the R bus. The test will pass if the selected byte has ones at all bit locations specified by the ones of the mask. Bytes are selected by programming the $\overline{SIO}$ inputs low. Test results are indicated on the ZERO output, which goes to one if the test passes. Register write is internally disabled for this instruction.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
No	No	No	Yes

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	Yes	Byte Select
$\overline{SIO1}$	Yes	Byte Select
$\overline{SIO2}$	Yes	Byte Select
$\overline{SIO3}$	Yes	Byte Select
Cn	No	Inactive

Status Signals

ZERO	=	1 if result (selected bytes) = Pass
N	=	0
OVR	=	0
C	=	0

EXAMPLE (assumes a 32-bit configuration)

Test bits 7, 6 and 5 of bytes 1 and 2 of data in register 3 for ones.

Instr Code I7-I0	Mask (LSH) A3-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Mask (MSH) C3-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$- $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0	$\overline{SIO3}$- $\overline{SIO0}$	$\overline{IESIO3}$- $\overline{IESIO0}$
0010 1000	0000	00 0011	X 00	1110	X	XXXX	XX	X	X	XXXX	0	X	110	1001	0000

Assume register file 3 holds 881CF003 (Hex).

Mask: 1110 0000 1110 0000 1110 0000 1110 0000 — Rn ← Mask (C3-C0::A3-A0)

Source: 1000 1000 0001 1100 1101 0000 0000 0011 — Sn ← RF(3)n†

Output: 0 — ZERO ← 0

†n = nth byte

FUNCTION

Performs one of N-2 iterations of nonrestoring unsigned division by a test subtraction of the N-bit divisor from the 2N-bit dividend. An algorithm using this instruction can be found in the "Other Arithmetic Instructions" section.

DESCRIPTION

UDIVI performs a test subtraction of the divisor from the dividend to generate a quotient bit. The test subtraction may pass or fail and is corrected in the subsequent instruction if it fails. Similarly a failed test from the previous instruction is corrected during evaluation of the current UDIVI instruction (see the "Other Arithmetic Instructions"section for more details).

The R bus must be loaded with the divisor, the S bus with the most significant half of the result of the previous instruction (UDIVI during iteration or UDIVIS at the beginning of iteration). The least significant half of the previous result is in the MQ register.

UDIVI checks the result of the previous pass/fail test and then evaluates:

$F \leftarrow R + S$ if the test is failed
$F \leftarrow R' + S + Cn$ if the test is passed

A double precision left shift is performed; bit 7 of the most significant byte of the MQ shifter is transferred to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte of the ALU shifter is lost. The unfixed quotient bit is circulated into the least significant bit of the MQ shifter.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes internally generated end-fill bit.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed high.

Status Signals

ZERO = 1 if result = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Computes the first quotient bit of nonrestoring unsigned division. An algorithm using this instruction is given in the "Other Arithmetic Instructjions" section.

DESCRIPTION

UDIVIS computes the first quotient bit during nonrestoring unsigned division by subtracting the divisor from the dividend. The resulting remainder due to subtraction may be negative; the subsequent UDIVI instruction may have to restore the remainder during the next operation.

The R bus must be loaded with the divisor and the S bus with the most significant half of the remainder. The result on the Y bus should be loaded back into the register file for use in the next instruction. The least significant half of the remainder is in the MQ register.

UDIVIS computes:

$$F \leftarrow R' + S + Cn$$

A double precision left shift is performed; bit 7 of the most significant byte of the MQ shifter is transferred to bit 0 of the least significant byte of the ALU shifter. Bit 7 of the most significant byte of the ALU shifter is lost. The unfixed quotient bit is circulated into the least significant bit of the MQ shifter.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Left	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes internally generated end-fill bit.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed high.

Status Signals

ZERO = 1 if intermediate result = 0
N = 0
OVR = 1 if divide overflow
C = 1 if carry-out

FUNCTION

Solves the final quotient bit during nonrestoring unsigned division. An algorithm using this instruction is given in the "Other Arithmetic Instructions" section.

DESCRIPTION

UDIVIT performs the final subtraction of the divisor from the remainder during nonrestoring signed division. UDIVIT is preceded by N-1 iterations of UDIVI, where N is the number of bits in the dividend.

The R bus must be loaded with the divisor, the S bus must be loaded with the most significant half of the result of the last UDIVI instruction. The least significant half lies in the MQ register. The Y bus result must be loaded back into the register file for use in the subsequent DIVRF instruction.

UDIVIT checks the results of the previous pass/fail test and evaluates:

$Y \leftarrow R + S$ if the test is failed
$Y \leftarrow R' + S + Cn$ if the test is passed

The contents of the MQ register are shifted one bit to the left; the unfixed quotient bit is circulated into the least significant bit.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
None	Left

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Inactive
$\overline{SIO0}$	No	Passes internally generated end-fill bit.
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
SIO3	No	
Cn	Yes	Should be programmed high.

Status Signals

ZERO = 1 if intermediate result = 0
N = 0
OVR = 0
C = 1 if carry-out

FUNCTION

Performs one of N unsigned multiplication iterations for computing an N-bit by N-bit product. An algorithm for unsigned multiplication using this instruction is given in the "Other Arithmetic Instructions" section.

DESCRIPTION

UMULI checks to determine whether the multiplicand should be added with the present partial product. The instruction evaluates:

F ← R + S + Cn	if the addition is required
F ← S	if no addition is required

A double precision right shift is performed. Bit 0 of the least significant byte of the ALU shifter is passed to bit 7 of the most significant byte of the MQ shifter; carry-out is passed to the most significant bit of the ALU shifter.

The S bus should be loaded with the contents of an accumulator and the R bus with the multiplicand. The Y bus result should be written back to the accumulator after each iteration of UMULI. The accumulator should be cleared and the MQ register loaded with the multiplier before the first iteration.

R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Recommended S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	No

Recommended Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port
Yes	No	Yes

Shift Operations

ALU	MQ
Right	Right

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Holds LSB of MQ.
$\overline{SIO0}$	No	Passes internal input (shifted bit).
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	Yes	Should be programmed low.

Status Signals†

ZERO = 1 if result = 0
N = 1 if MSB = 1
OVR = 0
C = 1 if carry-out

†Valid only on final execution of multiply iteration

FUNCTION

Evaluates the logical expression R XOR S.

DESCRIPTION

Data on the R bus is exclusive ORed with data on the S bus. The result appears at the ALU and MQ shifters.

*The result of this instruction can be shifted in the same microcycle by specifying a shift instruction in the upper nibble (I7-I4) of the instruction field. The result may also be passed without shift. Possible instructions are listed in Table 15.

Available R Bus Source Operands

RF (A5-A0)	A3-A0 Immed	DA-Port	C3-C0 :: A3-A0 Mask
Yes	No	Yes	No

Available S Bus Source Operands

RF (B5-B0)	DB-Port	MQ Register
Yes	Yes	Yes

Available Destination Operands

RF (C5-C0)	RF (B5-B0)	Y-Port	ALU Shifter	MQ Shifter
Yes	No	Yes	Yes	Yes

Control/Data Signals

Signal	User Programmable	Use
SSF	No	Affect shift instructions programmed in bits I7-I4 of instruction field.
$\overline{SIO0}$	No	
$\overline{SIO1}$	No	
$\overline{SIO2}$	No	
$\overline{SIO3}$	No	
Cn	No	Inactive

Status Signals†

ZERO	= 1 if result = 0
N	= 1 if MSB = 1
OVR	= 0
C	= 0

†C is ALU carry-out and is evaluated before shift operation. ZERO and N (negative) are evaluated after shift operation. OVR (overflow) is evaluated after ALU operation and after shift operation.

EXAMPLE (assumes a 32-bit configuration)

Exclusive OR the contents of register 3 and register 5, and store the result in register 5.

Instr Code I7-I0	Oprd Addr A5-A0	Oprd Addr B5-B0	Oprd Sel EB1- $\overline{EA}$ EB0	Dest Addr C5-C0	Destination Selects SELMQ	$\overline{WE3}$- $\overline{WE0}$	SELRF1- SELRF0	$\overline{OEA}$	$\overline{OEB}$	$\overline{OEY3}$ $\overline{OEY0}$	$\overline{OES}$	Cn	CF2- CF0
1111 1001	00 0011	00 0101	0 00	00 0101	0	0000	10	X	X	XXXX	0	X	110

Assume register file 3 holds 33F6D840 (Hex) and register file 5 holds 90F6D842 (Hex)..

Source 0011 0011 1111 0110 1101 1000 0100 0000 R ← RF(3)

Source 1001 0000 1111 0110 1101 1000 0100 0010 S ← RF(5)

Destination 1010 0011 0000 0000 0000 0000 0000 0010 RF(5) ← R XOR S