Z8681/82 Z8® ROMless MCU

FEATURES

- Complete microcomputer, 24 I/O lines, and up to 64K bytes of addressable external space each for program and data memory.

- 143-byte register file, including 124 general-purpose registers, 3 I/O port registers, and 16 status and control registers.

- Vectored, priority interrupts for I/O, counter/timers, and UART.

- On-chip oscillator that accepts crystal or external clock drive.

- Full-duplex UART and two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler.

- Register Pointer so that short, fast instructions can access any one of the nine working-register groups.

- Single + 5V power supply—all I/O pins TTL compatible.

- **Z8681/82 available in 8 MHz. Z8681 also available in 12 MHz.**

GENERAL DESCRIPTION

The Z8681 and Z8682 are ROMless versions of the Z8 single-chip microcomputer. The Z8682 is usually more cost effective. These products differ only slightly and can be used interchangeably with proper system design to provide maximum flexibility in meeting price and delivery needs.

The Z8681/82 offers all the outstanding features of the Z8 family architecture except an on-chip program ROM. Use of external memory rather than a preprogrammed ROM enables this Z8 microcomputer to be used in low volume applications or where code flexibility is required.
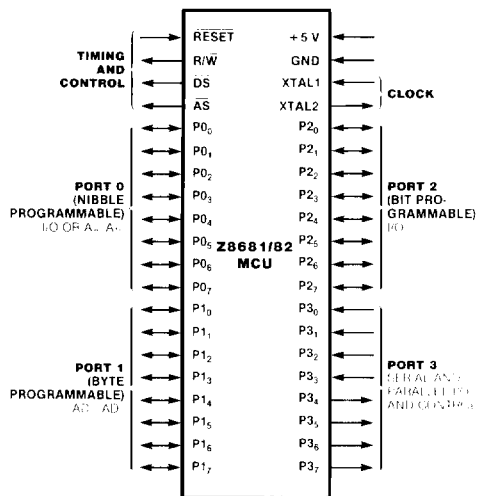

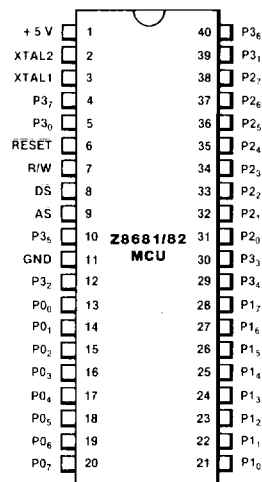
Figure 1. Pin Functions

Figure 2a. 40-pin Dual-In-Line Package (DIP), Pin Assignments

The Z8681/82 can provide up to 16 output address lines, thus permitting an address space of up to 64K bytes of data or program memory. Eight address outputs (AD_0-AD_7) are provided by a multiplexed, 8-bit, Address/Data bus. The remaining 8 bits can be provided by the software configuration of Port 0 to output address bits A_8-A_{15}.

Available address space can be doubled (up to 128K bytes for the Z8681 and 124K bytes for the Z8682) by programming bit 4 of Port 3 ($P3_4$) to act as a data memory select output ($\overline{DM}$). The two states of $\overline{DM}$ together with the 16 address outputs can define separate data and memory address spaces of up to 64K/62Kbytes each.

There are 143 bytes of RAM located on-chip and organized as a register file of 124 general-purpose registers, 16 control and status registers, and three I/O port registers. This register file can be divided into nine groups of 16 working registers each. Configuring the register file in this manner allows the use of short format instructions; in addition, any of the individual registers can be accessed directly.

The pin functions and the pin assignments of the Z8681/82 40— and 44—pin packages are illustrated in Figures 1 and 2, respectively.
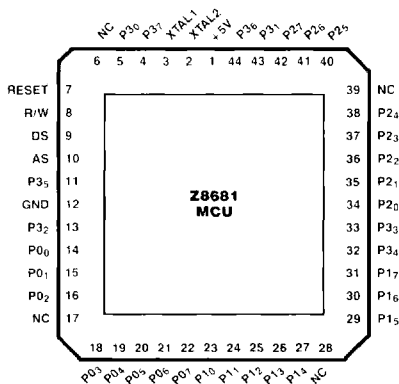

**Figure 2b. 44-pin Chip Carrier,
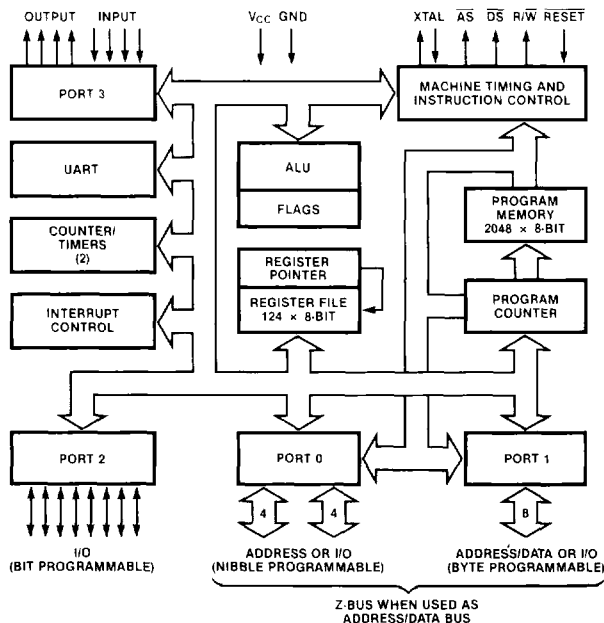Pin Assignments**

Figure 3. Functional Block Diagram

ARCHITECTURE

Z8681/82 architecture is characterized by a flexible I/O scheme, an efficient register and address space structure and a number of ancillary features that are helpful in many applications.

Microcomputer applications demand powerful I/O capabilities. The Z8681/82 fulfills this with 24 pins available for input and output. These lines are grouped into three ports of eight lines each and are configurable under software control to provide timing, status signals, serial or parallel I/O with or without handshake, and an Address bus for interfacing external memory.

Three basic address spaces are available: program memory, data memory and the register file (internal). The 143-byte random-access register file is composed of 124 general-purpose registers, three I/O port registers, and 16 control and status registers.

To unburden the program from coping with real-time problems such as serial data communication and counting/timing, an asynchronous receiver/transmitter (UART) and two counter/timers with a large number of user-selectable modes are offered on-chip. Hardware support for the UART is minimized because one of the on-chip timers supplies the bit rate. Figure 3 shows the Z8681/82 block diagram.

PIN DESCRIPTION

$\overline{AS}$. *Address Strobe* (output, active Low). Address Strobe is pulsed once at the beginning of each machine cycle. Addresses output via Port 1 for all external program or data memory transfers are valid at the trailing edge of $\overline{AS}$.

$\overline{DS}$. *Data Strobe* (output, active Low). Data Strobe is activated once for each external memory transfer.

P0$_0$-P0$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$. *I/O Port Lines* (input/outputs, TTL-compatible). These 24 lines are divided into three 8-bit I/O ports that can be configured under program control for I/O or external memory interface (Figure 3)

P1$_0$-P1$_7$. *Address/Data Port* (bidirectional). Multiplexed address (A$_0$-A$_7$) and data (D$_0$-D$_7$) lines used to interface with program and data memory.

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ initializes the Z8681/82. After RESET the Z8681 is in the extended memory mode. When $\overline{RESET}$ is deactivated, program execution begins from program location 000C$_H$ for the Z8681 and 0812$_H$ for the Z8682.

R/$\overline{W}$. *Read/Write* (output). R/$\overline{W}$ is Low when the Z8681/82 is writing to external program or data memory.

XTAL1, XTAL2. *Crystal 1, Crystal 2* (time-base input and output). These pins connect a parallel-resonant crystal to the on-chip clock oscillator and buffer.

SUMMARY OF Z8681 AND Z8682 DIFFERENCES

Feature	Z8681	Z8682
Address of first instruction executed after Reset	12	2066
Addressable memory space	0–64K	2K–64K
Address of interrupt vectors	0–11	2048–2065
Reset input high voltage	TTL levels *	7.35–8.0V
Port 0 configuration after Reset	Input, float after reset. Can be programmed as Address bits.	Output, configured as Address bit A$_8$–A$_{15}$.
External memory timing start-up configurations	Extended Timing	Normal Timing
Interrupt vectors	2 byte vectors point directly to service routines.	2 byte vectors in internal ROM point to 3 byte Jump instructions, which point to service routines.
Interrupt response time	26 clocks	36 clocks

* 8.0V V$_{IN}$ max.

ADDRESS SPACES

Program Memory*. The Z8681/82 addresses 64K/62K bytes of external program memory space (Figure 4).

For the Z8681, the first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Program execution begins at location 000C$_H$ after a reset.

The Z8682 has six 24-bit interrupt vectors beginning at address 0800$_H$. The vectors consist of Jump Absolute instructions. After a reset, program execution begins at location 0812$_H$ for the Z8682.

Data Memory*. The Z8681/82 can address 64K/62K bytes of external data memory. External data memory may be included with or separated from the external program memory space. $\overline{DM}$, an optional I/O function that can be programmed to appear on pin P3$_4$, is used to distinguish between data and program memory space.

Register File. The 143-byte register file includes three I/O port registers (R0, R2, R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255). These registers are assigned the address locations shown in Figure 5.

Z8681/82 instructions can access registers directly or indirectly with an 8-bit address field. This also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In the 4-bit mode, the register file is divided into nine working-register groups, each occupying 16 contiguous locations (Figure 5). The Register Pointer addresses the starting location of the active working-register group (Figure 6).

Stacks. Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4-R127).
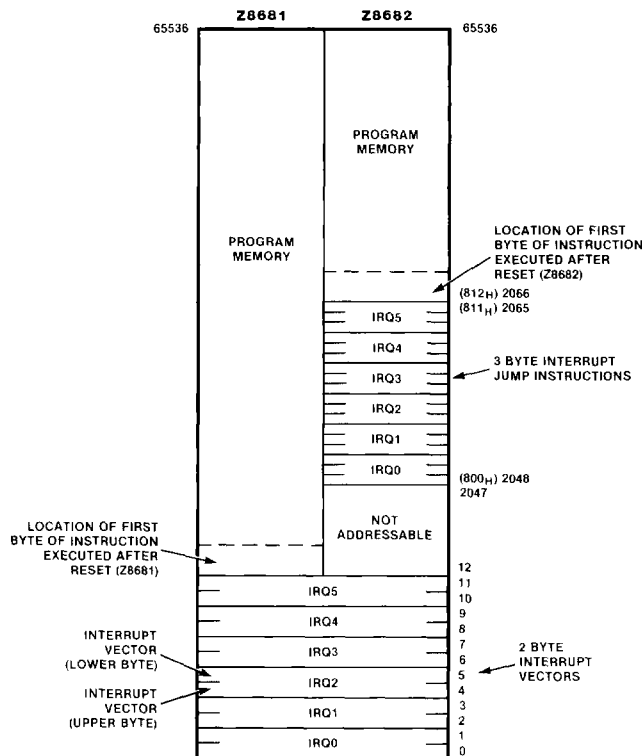

Figure 4. Z8681/82 Program Memory Map

DEC		HEX	IDENTIFIERS
255	STACK POINTER (BITS 7-0)	FF	SPL
254	STACK POINTER (BITS 15-8)	FE	SPH
253	REGISTER POINTER	FD	RP
252	PROGRAM CONTROL FLAGS	FC	FLAGS
251	INTERRUPT MASK REGISTER	FB	IMR
250	INTERRUPT REQUEST REGISTER	FA	IRQ
249	INTERRUPT PRIORITY REGISTER	F9	IPR
248	PORTS 0-1 MODE	F8	P01M
247	PORT 3 MODE	F7	P3M
246	PORT 2 MODE	F6	P2M
245	T0 PRESCALER	F5	PRE0
244	TIMER/COUNTER 0	F4	T0
243	T1 PRESCALER	F3	PRE1
242	TIMER/COUNTER 1	F2	T1
241	TIMER MODE	F1	TMR
240	SERIAL I/O	F0	SIO
	NOT IMPLEMENTED		
127		7F	
	GENERAL-PURPOSE REGISTERS		
4		04	
3	PORT 3	03	P3
2	PORT 2	02	P2
1	PORT 1	01	P1
0	PORT 0	00	P0

Figure 5. The Register File

THE UPPER NIBBLE OF THE REGISTER FILE ADDRESS PROVIDED BY THE REGISTER POINTER SPECIFIES THE ACTIVE WORKING-REGISTER GROUP.

SPECIFIED WORKING-REGISTER GROUP

THE LOWER NIBBLE OF THE REGISTER FILE ADDRESS PROVIDED BY THE INSTRUCTION POINTS TO THE SPECIFIED REGISTER.

I/O PORTS

Figure 6. The Register Pointer

SERIAL INPUT/OUTPUT

Port 3 lines $P3_0$ and $P3_7$ can be programmed as serial I/O lines for full-duplex serial asynchronous receiver/transmitter operation. The bit rate is controlled by Counter/Timer 0.

The Z8681/82 automatically adds a start bit and two stop bits to transmitted data (Figure 7). Odd parity is also available as an option. Eight data bits are always transmitted, regardless of parity selection. If parity is enabled, the eighth data bit is used as the odd parity bit. An interrupt request (IRQ4) is generated on all transmitted characters.

Received data must have a start bit, eight data bits, and at least one stop bit. If parity is on, bit 7 of the received data is replaced by a parity error flag. Received characters generate the IRQ3 interrupt request.

Transmitted Data (No Parity)

Received Data (No Parity)

Transmitted Data (With Parity)

Received Data (With Parity)

Figure 7. Serial Data Formats

COUNTER/TIMERS

The Z8681/82 contains two 8-bit programmable counter/timers (T_0 and T_1), each driven by its own 6-bit programmable prescaler. The T_1 prescaler can be driven by internal or external clock sources; however, the T_0 prescaler is driven by the internal clock only.

The 6-bit prescalers can divide the input frequency of the clock source by any number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of count, a timer interrupt request—IRQ4 (T_0) or IRQ5 (T_1)—is generated.

The counters can be started, stopped, restarted to continue, or restarted from the initial value. The counters can also be programmed to stop upon reaching zero (single-pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode). The counters, but not the prescalers, can be read any time without disturbing their value or count mode.

The clock source for T_1 is user-definable; it can be either the internal microprocessor clock divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input as an external clock, a trigger input that can be retriggerable or nonretriggerable, or as a gate input for the internal clock. The counter/timers can be programmably cascaded by connecting the T_0 output to the input of T_1. Port 3 line $P3_6$ also serves as a timer output (T_{OUT}) through which T_0, T_1 or the internal clock can be output.

I/O PORTS

The Z8681/82 has 24 lines available for input and output. These lines are grouped into three ports of eight lines each and are configurable as input, output or address. Under software control, the ports can be programmed to provide address outputs, timing, status signals, serial I/O, and parallel I/O with or without handshake. All ports have active pull-ups and pull-downs compatible with TTL loads.

Port 1 is a dedicated Z-BUS compatible memory interface. The operations of Port 1 are supported by the Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) lines, and by the Read/Write (R/$\overline{W}$) and Data Memory ($\overline{DM}$) control lines. The low-order program and data memory addresses (A_0-A_7) are output through Port 1 (Figure 8) and are multiplexed with data in/out (D_0-D_7). Instruction fetch and data memory read/write operations are done through this port.

Port 1 cannot be used as a register nor can a handshake mode be used with this port.

Both the Z8681 and Z8682 wake up with the 8 bits of Port 1 configured as address outputs for external memory. If more than eight address lines are required with the Z8681, additional lines can be obtained by programming Port 0 bits as address bits. The least-significant four bits of Port 0 can be configured to supply address bits A_8-A_{11} for 4K byte addressing or both nibbles of Port 0 can be configured to supply address bits A_8-A_{15} for 64K byte addressing.
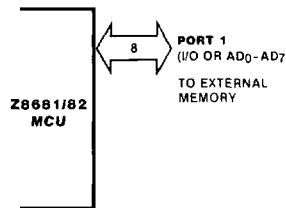

Figure 8. Port 1

Port 0* can be programmed as a nibble I/O port, or as an address port for interfacing external memory (Figure 9). When used as an I/O port, Port 0 can be placed under handshake control. In this configuration, Port 3 lines $P3_2$ and $P3_5$ are used as the handshake controls DAV_0 and RDY_0. Handshake signal assignment is dictated by the I/O direction of the upper nibble $P0_4$-$P0_7$.

For external memory references, Port 0 can provide address bits A_8-A_{11} (lower nibble) or A_8-A_{15} (lower and upper nibbles) depending on the required address space. If the address range requires 12 bits or less, the upper nibble of Port 0 can be programmed independently as I/O while the lower nibble is used for addressing.

In the Z8681*, Port 0 lines float after reset; their logic state is unknown until the execution of an initialization routine that configures Port 0.

* This feature differs in the Z8681 and Z8682.

Such an initialization routine must reside within the first 256 bytes of executable code and must be physically mapped into memory by forcing the Port 0 address lines to a known state (Figure 10). The proper port initialization sequence is:

1. Write initial address (A_8-A_{15}) of initialization routine to Port 0 address lines.

2. Configure Port 0 Mode register to output A_8-A_{15} (or A_8-A_{11}).

To permit the use of slow memory, an automatic wait mode of two oscillator clock cycles is configured for the bus timing of the Z8681 after each reset. The initialization routine could include reconfiguration to eliminate this extended timing mode.

The following example illustrates the manner in which an initialization routine can be mapped in a Z8681 system with 4K of memory.

Example. In Figure 10, the initialization routine is mapped to the first 256 bytes of program memory. Pull-down resistors maintain the address lines at a logic 0 level when these lines are floating. The leakage current caused by fanout must be taken into consideration when selecting the value of the pulldown resistors. The resistor value must be large enough to allow the Port 0 output driver to pull the line to a logic 1. Generally, pulldown resistors are incompatible with TTL loads. If Port 0 drives into TTL input loads ($I_{LOW} = 1.6$ mA) the external resistors should be tied to V_{CC} and the initialization routine put in address space $FF00_H$–$FFFF_H$.

In the Z8682*, Port 0 lines are configured as address lines A_8-A_{15} after a Reset. If one or both nibbles are needed for

I/O operation, they must be configured by writing to the Port 0 Mode register. The Z8682 is in the fast memory timing mode after Reset, so the initialization routine must be in fast memory.
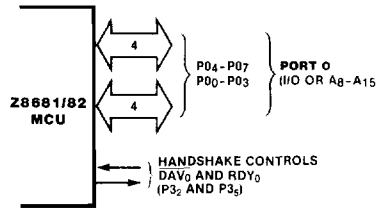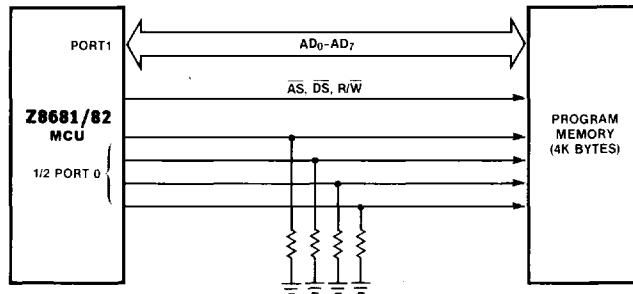

Figure 9. Port 0

Figure 10. Port 0 Address Lines Tied to Logic 0

Port 2 bits can be programmed independently as input or output (Figure 11). This port is always available for I/O operations. In addition, Port 2 can be configured to provide open-drain outputs.

Like Port 0, Port 2 may also be placed under handshake control. In this configuration, Port 3 lines $P3_1$ and $P3_6$ are used as the handshake controls lines $\overline{DAV}_2$ and RDY_2. The handshake signal assignment for Port 3 lines $P3_1$ and $P3_6$ is dictated by the direction (input or output) assigned to bit 7 of Port 2.
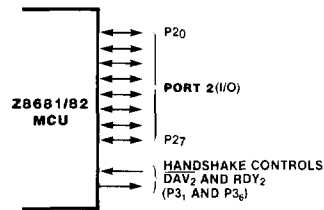
Figure 11. Port 2

Port 3 lines can be configured as I/O or control lines (Figure 12). In either case, the direction of the eight lines is fixed as four input ($P3_0$-$P3_3$) and four output ($P3_4$-$P3_7$). For serial I/O, lines $P3_0$ and $P3_7$ are programmed as serial in and serial out, respectively.

Port 3 can also provide the following control functions: handshake for Ports 0 and 2 ($\overline{DAV}$ and RDY); four external interrupt request signals (IRQ0-IRQ3); timer input and output signals (T_{IN} and T_{OUT}) and Data Memory Select ($\overline{DM}$).
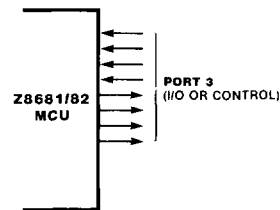
Figure 12. Port 3

*This feature differs in the Z8681 and Z8682.

INTERRUPTS*

The Z8681/82 allows six different interrupts from eight sources: the four Port 3 lines $P3_0$-$P3_3$, Serial In, Serial Out, and the two counter/timers. These interrupts are both maskable and prioritized. The Interrupt Mask register globally or individually enables or disables the six interrupt requests. When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register.

All Z8681 and Z8682 interrupts are vectored through locations in program memory. When an interrupt request is granted, an interrupt machine cycle is entered. This disables all subsequent interrupts, saves the Program Counter and status flags, and accesses the program memory vector location reserved for that interrupt. In the Z8681, this memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request. The Z8681 takes 63 crystal cycles to enter an interrupt subroutine.

The Z8682 has a small internal ROM that contains six 2-byte interrupt vectors pointing to addresses 2048-2065, where 3-byte jump absolute instructions are located (Figure 4 and Table 1). These jump instructions each contain a 1-byte opcode and a 2-byte starting address for the interrupt service routine.

Table 1. Z8682 Interrupt Processing

Hex Address	Contains Jump Instruction and Subroutine Address For
800-802	IRQ0
803-805	IRQ1
806-808	IRQ2
809-80B	IRQ3
80C-80E	IRQ4
80F-811	IRQ5

Polled interrupt systems are also supported. To accommodate a polled structure, any or all of the interrupt inputs can be masked and the Interrupt Request register polled to determine which of the interrupt requests needs service.

CLOCK

The on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal or to any suitable external clock source (XTAL1 = Input, XTAL2 = Output).

The crystal source is connected across XTAL1 and XTAL2, using the recommended capacitance (C_L = 15 pf maximum) from each pin to ground. The specifications for the crystal are as follows:

- AT cut, parallel-resonant
- Fundamental type
- Series resistance, $R_s \leqslant 100\Omega$
- For Z8682, 8 MHz maximum
- **For Z8681–12, 16 MHz maximum**

Z8681/Z8682 INTERCHANGEABILITY

Although the Z8681 and Z8682 have minor differences, a system can be designed for compatibility with both ROMless versions. To achieve interchangeability, the design must take into account the special requirements of each device in the external interface, initialization, and memory mapping.

External Interface. The Z8682 requires a 7.5V positive logic level on the RESET pin for at least 6 clock periods immediately following reset, as shown in Figure 13. The Z8681 requires a 3.8V or higher positive logic level, but is compatible with the Z8682 RESET waveform. Figure 14 shows a simple circuit for generating the 7.5V level.
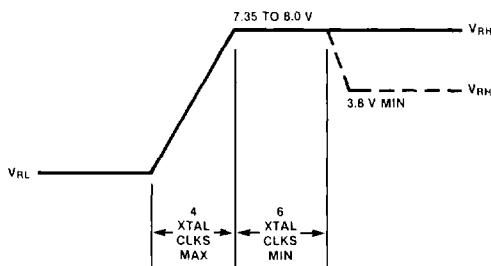
Figure 13. Z8682 RESET Pin Input Waveform
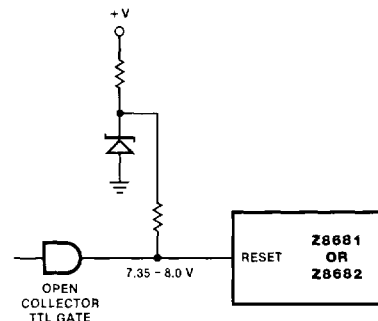
Figure 14. RESET Circuit

Initialization. The Z8681 wakes up after reset with Port 0 configured as an input, which means Port 0 lines are floating in a high-impedance state. Because of this pullup or pulldown, resistors must be attached to Port 0 lines to force them to a valid logic level until Port 0 is configured as an address port.

Port 0 initialization is discussed in the section on ports. An example of an initialization routine for Z8681/Z8682 compatibility is shown in Table 2. Only the Z8681 need execute this program.

Table 2. Initialization Routine

Address	Opcodes	Instruction	Comments
000C	E6 00 00	LD P0 #%00	Set A_8-A_{15} to 0.
000F	E6 F8 96	LD P01M #%96	Configure Port 0 as A_8-A_{15}. Eliminate extended memory timing.
0012	8D 08 12	JP START ADDRESS	Execute application program.
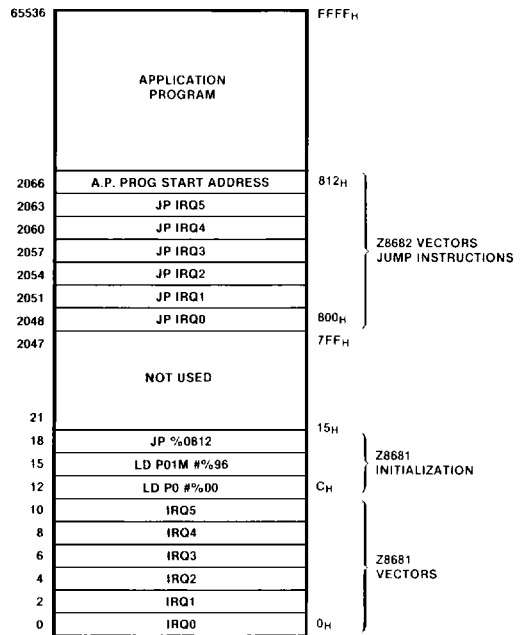
Figure 15. Z8681/82 Logical Program Memory Mapping

Memory Mapping. The Z8681 and Z8682 lower memory boundaries are located at 0 and 2048, respectively. A single program ROM can be used with either product if the logical program memory map shown in Figure 15 is followed. The Z8681 vectors and initialization routine must be starting at address 0 and the Z8682 3-byte vectors (jump instructions) must be at address 2048 and higher. Addresses in the range 21-2047 are not used. Figure 16 shows practical schemes for implementing this memory map using 4K and 2K ROMs.

a. Logical to Physical Memory Mapping for 4K ROM

b. Logical to Physical Memory Mapping for 2K ROM
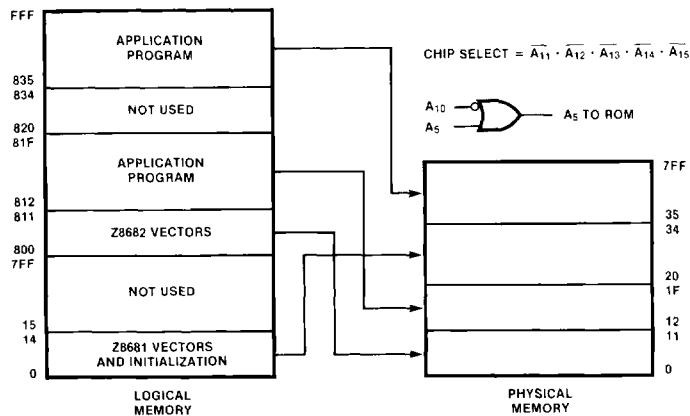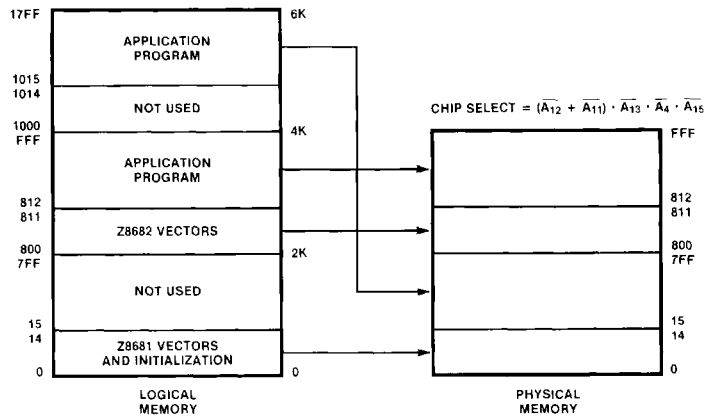
Figure 16. Practical Schemes for Implementing Z8681 and Z8682 Compatible Memory Map

INSTRUCTION SET NOTATION

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate
R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working register pair address

Symbols. The following symbols are used in describing the instruction set.

dst	Destination location or contents
src	Source location or contents
cc	Condition code (see list)
@	Indirect address prefix
SP	Stack pointer (control registers 254-255)
PC	Program counter
FLAGS	Flag register (control register 252)
RP	Register pointer (control register 253)
IMR	Interrupt mask register (control register 251)

Assignment of a value is indicated by the symbol "←". For example,

$$dst \leftarrow dst + src$$

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr(n)" is used to refer to bit "n" of a given location. For example,

$$dst\,(7)$$

refers to bit 7 of the destination operand.

Flags. Control Register R252 contains the following six flags:

C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag
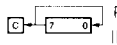
Affected flags are indicated by:
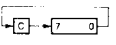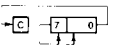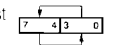
0	Cleared to zero
1	Set to one
*	Set or cleared according to operation
—	Unaffected
X	Undefined

CONDITION CODES

Value	Mnemonic	Meaning	Flags Set
1000		Always true	—
0111	C	Carry	C = 1
1111	NC	No carry	C = 0
0110	Z	Zero	Z = 1
1110	NZ	Not zero	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No overflow	V = 0
0110	EQ	Equal	Z = 1
1110	NE	Not equal	Z = 0
1001	GE	Greater than or equal	(S XOR V) = 0
0001	LT	Less than	(S XOR V) = 1
1010	GT	Greater than	[Z OR (S XOR V)] = 0
0010	LE	Less than or equal	[Z OR (S XOR V)] = 1
1111	UGE	Unsigned greater than or equal	C = 0
0111	ULT	Unsigned less than	C = 1
1011	UGT	Unsigned greater than	(C = 0 AND Z = 0) = 1
0011	ULE	Unsigned less than or equal	(C OR Z) = 1
0000		Never true	—

INSTRUCTION FORMATS

```
┌──────────┐
│   OPC    │          CCF, DI, EI, IRET, NOP,
└──────────┘          RCF, RET, SCF

┌─────┬────┐
│ dst │ OPC│          INC r
└─────┴────┘
```

One-Byte Instruction

```
┌─────┬──────┐
│ OPC │ MODE │            CLR, CPL, DA, DEC,
├─────┴──────┤  OR ┌─────┬───────┐   DECW, INC, INCW, POP,
│  dst/src   │     │1 1 1 0│dst/src│   PUSH, RL, RLC, RR,
└────────────┘     └─────┴───────┘   RRC, SRA, SWAP

┌──────────┐
│   OPC    │            JP, CALL (Indirect)
├──────────┤  OR ┌─────┬─────┐
│   dst    │     │1 1 1 0│ dst │
└──────────┘     └─────┴─────┘

┌──────────┐
│   OPC    │            SRP
├──────────┤
│  VALUE   │
└──────────┘

┌─────┬──────┐
│ OPC │ MODE │            ADC, ADD, AND,
├─────┼──────┤            CP, OR, SBC, SUB,
│ dst │ src  │            TCM, TM, XOR
└─────┴──────┘

┌──────┬─────┐
│ MODE │ OPC │            LD, LDE, LDEI,
├──────┴─────┤            LDC, LDCI
│dst/src│src/dst│
└──────┴─────┘

┌───────┬─────┐
│dst/src│ OPC │            LD
├───────┴─────┤  OR ┌─────┬─────┐
│ src/dst     │     │1 1 1 0│ src │
└─────────────┘     └─────┴─────┘

┌─────┬─────┐
│ dst │ OPC │            LD
├─────┴─────┤
│  VALUE    │
└───────────┘

┌────────┬─────┐
│ dst/CC │ OPC │            DJNZ, JR
├────────┴─────┤
│     RA       │
└──────────────┘
```

Two-Byte Instruction

```
┌─────┬──────┐
│ OPC │ MODE │            ADC, ADD, AND, CP,
├─────┴──────┤  OR ┌─────┬─────┐   LD, OR, SBC, SUB,
│    src     │     │1 1 1 0│ src │   TCM, TM, XOR
├────────────┤  OR ┌─────┬─────┐
│    dst     │     │1 1 1 0│ dst │
└────────────┘     └─────┴─────┘

┌─────┬──────┐
│ OPC │ MODE │            ADC, ADD, AND, CP,
├─────┴──────┤  OR ┌─────┬─────┐   LD, OR, SBC, SUB,
│    dst     │     │1 1 1 0│ dst │   TCM, TM, XOR
├────────────┤
│   VALUE    │
└────────────┘

┌──────┬─────┐
│ MODE │ OPC │            LD
├──────┴─────┤  OR ┌─────┬─────┐
│    src     │     │1 1 1 0│ src │
├────────────┤  OR ┌─────┬─────┐
│    dst     │     │1 1 1 0│ dst │
└────────────┘     └─────┴─────┘

┌──────┬─────┐
│ MODE │ OPC │            LD
├──────┴─────┤
│dst/src│ x  │
├────────────┤
│  ADDRESS   │
└────────────┘

┌─────┬─────┐
│ cc  │ OPC │            JP
├─────┴─────┤
│   DAᵤ     │
├───────────┤
│   DAₗ     │
└───────────┘

┌───────────┐
│   OPC     │            CALL
├───────────┤
│   DAᵤ     │
├───────────┤
│   DAₗ     │
└───────────┘
```

Three-Byte Instruction

Figure 17. Instruction Formats

INSTRUCTION SUMMARY

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	C	Z	S	V	D	H
ADC dst,src dst ← dst + src + C	(Note 1)		1!.	*	*	*	*	0	*
ADD dst,src dst ← dst + src	(Note 1)		0:	*	*	*	*	0	*
AND dst,src dst ← dst AND src	(Note 1)		5 .	–	*	*	0	–	–
CALL dst SP ← SP – 2 @SP ← PC: PC ← dst	DA IRR		D6 D4	–	–	–	–	–	–
CCF C ← NOT C			EF	*	–	–	–	–	–
CLR dst dst ← 0	R IR		B0 B1	–	–	–	–	–	–
COM dst dst ← NOT dst	R IR		60 61	–	*	*	0	–	–
CP dst,src dst – src	(Note 1)		A:	*	*	*	*	–	–
DA dst dst ← DA dst	R IR		40 41	*	*	*	X	–	–

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	C	Z	S	V	D	H
DEC dst dst ← dst – 1	R IR		00 01	–	*	*	*	–	–
DECW dst dst ← dst – 1	RR IR		80 81	–	*	*	*	–	–
DI IMR (7) ← 0			8F	–	–	–	–	–	–
DJNZ r,dst r ← r – 1 if r ≠ 0 PC ← PC + dst Range: +127, –128	RA		rA r = 0 – F	–	–	–	–	–	–
EI IMR (7) ← 1			9F	–	–	–	–	–	–
INC dst dst ← dst + 1	r R IR		rE r = 0 – F 20 21	–	*	*	*	–	–
INCW dst dst ← dst + 1	RR IR		A0 A1	–	*	*	*	–	–

Instruction and Operation	Addr Mode dst	src	Opcode Byte (Hex)	C	Z	S	V	D	H
IRET FLAGS ← @SP; SP ← SP + 1 PC ← @SP; SP ← SP + 2; IMR (7) ← 1			BF	*	*	*	*	*	*
JP cc.dst if cc is true PC ← dst	DA IRR		cD c = 0 – F 30	—	—	—	—	—	—
JR cc.dst if cc is true, PC ← PC + dst Range: + 127, – 128	RA		cB c = 0 – F						
LD dst.src dst ← src	r r R	Im R r r = 0 – F	rC r8 r9	—	—	—	—	—	—
	r X r Ir R R IR IR	X r Ir r R IR IM R IM	C7 D7 E3 F3 E4 E5 E6 E7 F5						
LDC dst.src dst ← src	r Irr	Irr r	C2 D2						
LDCI dst.src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	C3 D3						
LDE dst.src dst ← src	r Irr	Irr r	82 92						
LDEI dst.src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	83 93						
NOP			FF						
OR dst.src dst ← dst OR src	(Note 1)		4*	—	*	*	0	—	—
POP dst dst ← @SP; SP ← SP + 1	R IR		50 51						
PUSH src SP ← SP – 1; @SP ← src	R IR		70 71						
RCF C ← 0			CF	0	—	—	—	—	—
RET PC ← @SP; SP ← SP + 2			AF						
RL dst	R IR		90 91	*	*	*	*	—	—

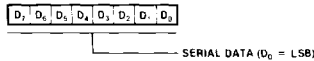| Instruction and Operation	Addr Mode dst	src	Opcode Byte (Hex)	C	Z	S	V	D	H
RLC dst	R IR		10 11	*	*	*	*	—	—
RR dst	R IR		E0 E1	*	*	*	*	—	—
RRC dst	R IR		C0 C1	*	*	*	*	—	—
SBC dst.src dst ← dst ← src ← C	(Note 1)		3*	*	*	*	*	1	*
SCF C ← 1			DF	1	—	—	—	—	—
SRA dst	R IR		D0 D1	*	*	*	0	—	—
SRP src RP ← src		Im	31	—	—	—	—	—	—
SUB dst.src dst ← dst ← src	(Note 1)		2*	*	*	*	*	1	*
SWAP dst	R IR		F0 F1	X	*	*	X	—	—
TCM dst.src (NOT dst) AND src	(Note 1)		6*	—	*	*	0	—	—
TM dst.src dst AND src	(Note 1)		7*	—	*	*	0	—	—
XOR dst.src dst ← dst XOR src	(Note 1)		B*	—	*	*	0	—	—

NOTE. These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a * in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.
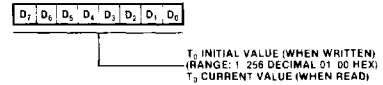
Addr Mode dst	src	Lower Opcode Nibble
r	r	[2]
r	Ir	[3]
R	R	[4]
R	IR	[5]
R	IM	[6]
IR	IM	[7]

REGISTERS

R240 SIO
Serial I/O Register
(F0$_H$; Read/Write)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

SERIAL DATA (D$_0$ = LSB)

R244 TO
Counter/Timer 0 Register
(F4$_H$; Read/Write)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

T$_0$ INITIAL VALUE (WHEN WRITTEN)
(RANGE: 1-256 DECIMAL 01-00 HEX)
T$_0$ CURRENT VALUE (WHEN READ)

R241 TMR
Time Mode Register
(F1$_H$; Read/Write)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

T$_{OUT}$ MODES
NOT USED = 00
T$_0$ OUT = 01
T$_1$ OUT = 10
INTERNAL CLOCK OUT = 11

T$_{IN}$ MODES
EXTERNAL CLOCK INPUT = 00
GATE INPUT = 01
TRIGGER INPUT = 10
(NON-RETRIGGERABLE)
TRIGGER INPUT = 11
(RETRIGGERABLE)

0 = NO FUNCTION
1 = LOAD T$_0$

0 = DISABLE T$_0$ COUNT
1 = ENABLE T$_0$ COUNT

0 = NO FUNCTION
1 = LOAD T$_1$

0 = DISABLE T$_1$ COUNT
1 = ENABLE T$_1$ COUNT
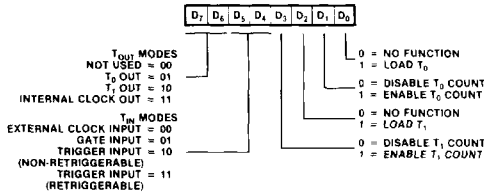
R245 PRE0
Prescaler 0 Register
(F5$_H$; Write Only)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

COUNT MODE
0 = T$_0$ SINGLE-PASS
1 = T$_0$ MODULO-N

RESERVED (MUST BE 0)

PRESCALER MODULO
(RANGE: 1-64 DECIMAL
01-00 HEX)
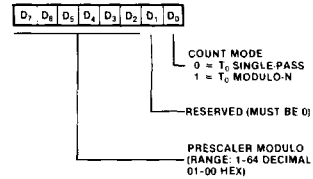
R242 T1
Counter Timer 1 Register
(F2$_H$; Read/Write)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

T$_1$ INITIAL VALUE (WHEN WRITTEN)
(RANGE 1-256 DECIMAL 01-00 HEX)
T$_1$ CURRENT VALUE (WHEN READ)
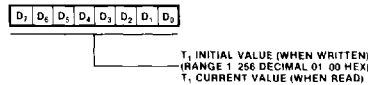
R246 P2M
Port 2 Mode Register
(F6$_H$; Write Only)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

P2$_0$-P2$_7$ I/O DEFINITION
0 DEFINES BIT AS OUTPUT
1 DEFINES BIT AS INPUT
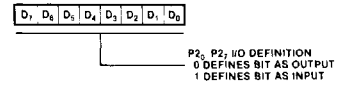
R243 PRE1
Prescaler 1 Register
(F3$_H$; Write Only)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

COUNT MODE
1 = T$_1$ MODULO-N
0 = T$_1$ SINGLE-PASS

CLOCK SOURCE
1 = T$_1$ INTERNAL
0 = T$_1$ EXTERNAL
TIMING INPUT
(T$_{IN}$) MODE

PRESCALER MODULO
(RANGE: 1-64 DECIMAL
01-00 HEX)
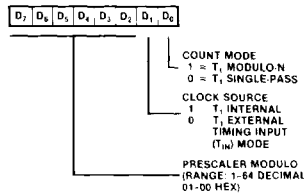
R247 P3M
Port 3 Mode Register
(F7$_H$; Write Only)

D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$

0 PORT 2 PULL-UPS OPEN DRAIN
1 PORT 2 PULL-UPS ACTIVE

RESERVED (MUST BE 0)

0 P3$_2$ = INPUT   P3$_5$ = OUTPUT
1 P3$_2$ = DAV0/RDY0 P3$_5$ = RDY0/DAV0

0 0 P3$_3$ = INPUT   P3$_4$ = OUTPUT
0 1
1 0 P3$_3$ = INPUT   P3$_4$ = DM
1 1 RESERVED

0 P3$_1$ = INPUT (T$_{IN}$) P3$_6$ = OUTPUT (T$_{OUT}$)
1 P3$_1$ = DAV2/RDY2 P3$_6$ = RDY2/DAV2

0 P3$_0$ = INPUT   P3$_7$ = OUTPUT
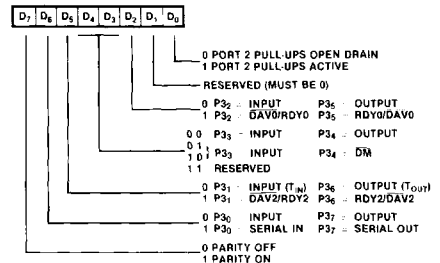1 P3$_0$ = SERIAL IN P3$_7$ = SERIAL OUT

0 PARITY OFF
1 PARITY ON

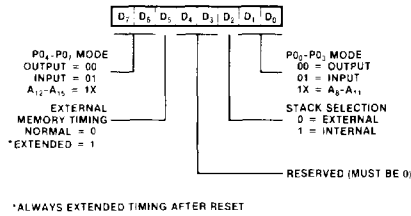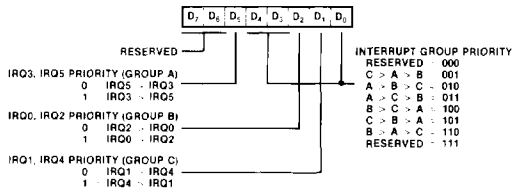**Figure 18. Control Registers**

REGISTERS
(Continued)

R248 P01M
Port 0 Register
(F8$_H$: Write Only)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

P0₄-P0₇ MODE
OUTPUT = 00
INPUT = 01
A₁₂-A₁₅ = 1X

EXTERNAL MEMORY TIMING
NORMAL = 0
*EXTENDED = 1

P0₀-P0₃ MODE
00 = OUTPUT
01 = INPUT
1X = A₈-A₁₁

STACK SELECTION
0 = EXTERNAL
1 = INTERNAL

RESERVED (MUST BE 0)

*ALWAYS EXTENDED TIMING AFTER RESET

R252 FLAGS
Flag Register
(FC$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

USER FLAG F1
USER FLAG F2
HALF CARRY FLAG
DECIMAL ADJUST FLAG
OVERFLOW FLAG
SIGN FLAG
ZERO FLAG
CARRY FLAG
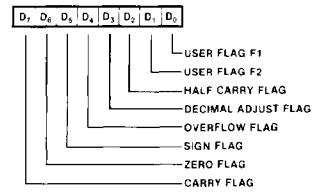
R249 IPR
Interrupt Priority Register
(F9$_H$: Write Only)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

RESERVED
IRQ3, IRQ5 PRIORITY (GROUP A)
0   IRQ5 > IRQ3
1   IRQ3 > IRQ5
IRQ0, IRQ2 PRIORITY (GROUP B)
0   IRQ2 > IRQ0
1   IRQ0 > IRQ2
IRQ1, IRQ4 PRIORITY (GROUP C)
0   IRQ1 > IRQ4
1   IRQ4 > IRQ1

INTERRUPT GROUP PRIORITY
RESERVED - 000
C > A > B   001
A > B > C - 010
A > C > B - 011
B > C > A - 100
C > B > A - 101
B > A > C - 110
RESERVED - 111

R253 RP
Register Pointer
(FD$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

REGISTER POINTER
r₇
r₆
r₅
r₄

DON'T CARE
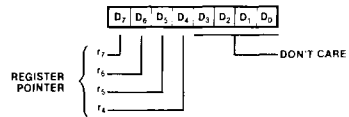
R250 IRQ
Interrupt Request Register
(FA$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

RESERVED (MUST BE 0)

IRQ0 = P3₂ INPUT (D₀ = IRQ0)
IRQ1 = P3₃ INPUT
IRQ2 = P3₁ INPUT
IRQ3 = P3₀ INPUT, SERIAL INPUT
IRQ4 = T₀, SERIAL OUTPUT
IRQ5 = T₁
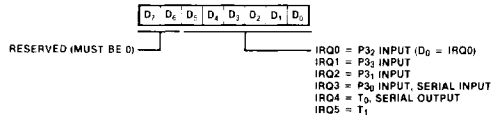
R254 SPH
Stack Pointer
(FE$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

STACK POINTER UPPER
BYTE (SP₈-SP₁₅)
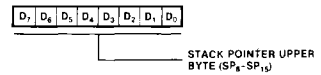
R251 IMR
Interrupt Mask Register
(FB$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

1 ENABLES IRQ0-IRQ5
(D₀ = IRQ0)
RESERVED (MUST BE 0)
1 ENABLES INTERRUPTS
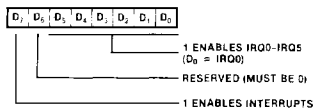
R255 SPL
Stack Pointer
(FF$_H$: Read/Write)

```
| D₇ | D₆ | D₅ | D₄ | D₃ | D₂ | D₁ | D₀ |
```

STACK POINTER LOWER
BYTE (SP₀-SP₇)
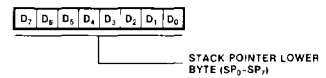
Figure 18. Control Registers (Continued)

Z8681/82 OPCODE MAP

Upper Nibble (Hex) — rows; Lower Nibble (Hex) — columns

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6.5 DEC R1	6.5 DEC IR1	6.5 ADD r1,r2	6.5 ADD r1,Ir2	10.5 ADD R2,R1	10.5 ADD IR2,R1	10.5 ADD R1,IM	10.5 ADD IR1,IM	6.5 LD r1,R2	6.5 LD r2,R1	12/10.5 DJNZ r1,RA	12/10.0 JR cc,RA	6.5 LD r1,IM	12/10.0 JP cc,DA	6.5 INC r1	
1	6.5 RLC R1	6.5 RLC IR1	6.5 ADC r1,r2	6.5 ADC r1,Ir2	10.5 ADC R2,R1	10.5 ADC IR2,R1	10.5 ADC R1,IM	10.5 ADC IR1,IM								
2	6.5 INC R1	6.5 INC IR1	6.5 SUB r1,r2	6.5 SUB r1,Ir2	10.5 SUB R2,R1	10.5 SUB IR2,R1	10.5 SUB R1,IM	10.5 SUB IR1,IM								
3	8.0 JP IRR1	6.1 SRP IM	6.5 SBC r1,r2	6.5 SBC r1,Ir2	10.5 SBC R2,R1	10.5 SBC IR2,R1	10.5 SBC R1,IM	10.5 SBC IR1,IM								
4	8.5 DA R1	8.5 DA IR1	6.5 OR r1,r2	6.5 OR r1,Ir2	10.5 OR R2,R1	10.5 OR IR2,R1	10.5 OR R1,IM	10.5 OR IR1,IM								
5	10.5 POP R1	10.5 POP IR1	6.5 AND r1,r2	6.5 AND r1,Ir2	10.5 AND R2,R1	10.5 AND IR2,R1	10.5 AND R1,IM	10.5 AND IR1,IM								
6	6.5 COM R1	6.5 COM IR1	6.5 TCM r1,r2	6.5 TCM r1,Ir2	10.5 TCM R2,R1	10.5 TCM IR2,R1	10.5 TCM R1,IM	10.5 TCM IR1,IM								
7	10/12.1 PUSH R2	12/14.1 PUSH IR2	6.5 TM r1,r2	6.5 TM r1,Ir2	10.5 TM R2,R1	10.5 TM IR2,R1	10.5 TM R1,IM	10.5 TM IR1,IM								
8	10.5 DECW RR1	10.5 DECW IR1	12.0 LDE r1,Irr2	18.0 LDEI Ir1,Irr2											6.1 DI	
9	6.5 RL R1	6.5 RL IR1	12.0 LDE r2,Irr1	18.0 LDEI Ir2,Irr1											6.1 EI	
A	10.5 INCW RR1	10.5 INCW IR1	6.5 CP r1,r2	6.5 CP r1,Ir2	10.5 CP R2,R1	10.5 CP IR2,R1	10.5 CP R1,IM	10.5 CP IR1,IM							14.0 RET	
B	6.5 CLR R1	6.5 CLR IR1	6.5 XOR r1,r2	6.5 XOR r1,Ir2	10.5 XOR R2,R1	10.5 XOR IR2,R1	10.5 XOR R1,IM	10.5 XOR IR1,IM							16.0 IRET	
C	6.5 RRC R1	6.5 RRC IR1	12.0 LDC r1,Irr2	18.0 LDCI Ir1,Irr2				10.5 LD r1,x,R2							6.5 RCF	
D	6.5 SRA R1	6.5 SRA IR1	12.0 LDC r2,Irr1	18.0 LDCI Ir2,Irr1	20.0 CALL* IRR1		20.0 CALL DA	10.5 LD r2,x,R1							6.5 SCF	
E	6.5 RR R1	6.5 RR IR1		6.5 LD r1,IR2	10.5 LD R2,R1	10.5 LD IR2,R1	10.5 LD R1,IM	10.5 LD IR1,IM							6.5 CCF	
F	8.5 SWAP R1	8.5 SWAP IR1		6.5 LD r1,r2		10.5 LD R2,R1									6.0 NOP	

Bytes per Instruction:
- Columns 0–3: **2**
- Columns 4–7: **3**
- Columns 8–B: **2**
- Columns C–D: **3**
- Column E: **1**

Legend example cell:

```
        LOWER
       OPCODE
       NIBBLE
          ↓
EXECUTION      PIPELINE
CYCLES   →  4  ←  CYCLES
           10.5
UPPER       CP   ← MNEMONIC
OPCODE  → A
NIBBLE     R2,R1
       ↙        ↖
   FIRST      SECOND
 OPERAND     OPERAND
```
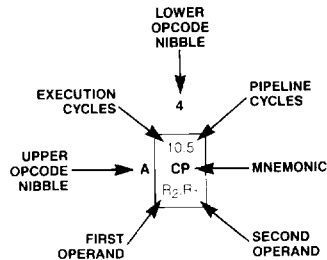
Legend:
R = 8-bit address
r = 4-bit address
R_1 or r_1 = Dst address
R_2 or r_2 = Src address

Sequence:
Opcode, First Operand, Second Operand

NOTE: The blank areas are not defined

*2-byte instruction, fetch cycle appears as a 3-byte instruction

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins except $\overline{RESET}$
   with respect to GND . . . . $-0.3V$ to $+7.0V$
Operating Ambient
   Temperature . . . . See Ordering Information
Storage Temperature . . . . $-65°C$ to $+150°C$

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

The DC characteristics listed below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

Standard conditions are as follows:

- $+4.75V \leqslant V_{CC} \leqslant +5.25V$

- $GND = 0V$

- $0°C \leqslant T_A \leqslant +70°C$ for S (Standard temperature)

- $-40°C \leqslant T_A \leqslant +100°C$ for E (Extended temperature)
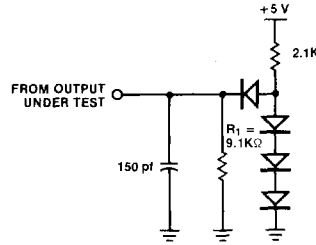
Figure 19. Test Load 1

DC CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit	Condition
V_{CH}	Clock Input High Voltage	3.8	V_{CC}	V	Driven by External Clock Generator
V_{CL}	Clock Input Low Voltage	-0.3	0.8	V	Driven by External Clock Generator
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{RH}	Reset Input High Voltage	3.8	V_{CC}	V	See Note
V_{RL}	Reset Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -250\,\mu A$
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = +2.0\,mA$
I_{IL}	Input Leakage	-10	10	μA	$0V \leqslant V_{IN} \leqslant +5.25V$
I_{OL}	Output Leakage	-10	10	μA	$0V \leqslant V_{IN} \leqslant +5.25V$
I_{IR}	Reset Input Current		-50	μA	$V_{CC} = +5.25V, V_{RL} = 0V$
I_{CC}	V_{CC} Supply Current		**150**	mA	All outputs and I/O pins floating

*The Reset line (pin 6) is used to place the Z8682 in external memory mode. This is accomplished as shown in Figure 13.
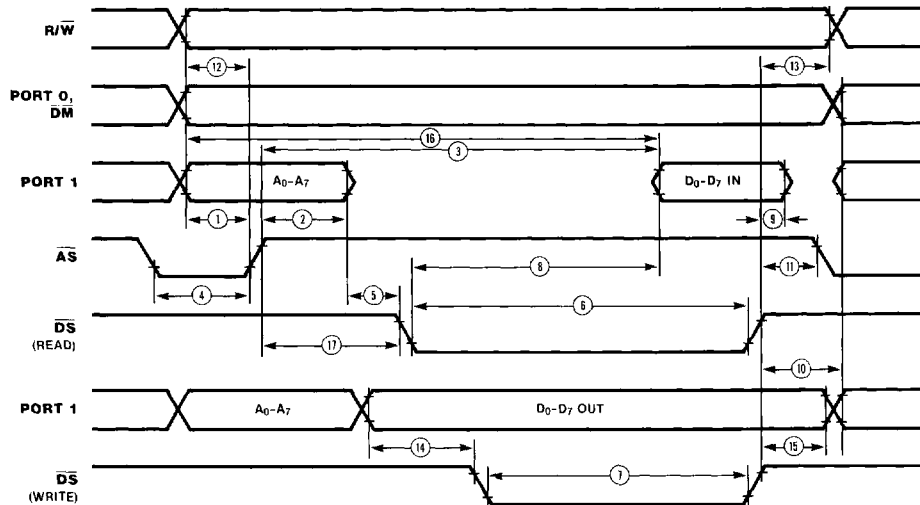
Figure 20. External I/O or Memory Read/Write Timing

AC CHARACTERISTICS

External I/O or Memory Read and Write Timing

Number	Symbol	Parameter	Z8681/82 8 MHz Min	Z8681/82 8 MHz Max	Z8681 12 MHz Min	Z8681 12 MHz Max	Z8681 16 MHz Min	Z8681 16 MHz Max	Notes
1	TdA(AS)	Address Valid to AS ↑Delay	50		35		20		2,3
2	TdAS(A)	AS ↑ to Address Float Delay	70		45		30		2,3
3	TdAS(DR)	AS ↑ to Read Data Required Valid		360		220		180	1,2,3
4	TwAS	AS Low Width	80		55		35		2,3
5	TdAz(DS)	Address Float to DS ↓	0		0		0		
6	TwDSR	DS (Read) Low Width	250		185		135		1,2,3
7	TwDSW	DS (Write) Low Width	160		110		80		1,2,3
8	TdDSR(DR)	DS ↓ to Read Data Required Valid		200		130		75	1,2,3
9	ThDR(DS)	Read Data to DS ↑ Hold Time	0		0		0		2,3
10	TdDS(A)	DS ↑ to Address Active Delay	70		45		-		2,3
11	TdDS(AS)	DS ↑ to AS ↓Delay	70		55		30		2,3
12	TdR/W(AS)	R/W Valid to AS ↑ Delay	50		30		20		2,3
13	TdDS(R/W)	DS ↑ to R/W Not Valid	60		35		30		2,3
14	TdDW(DSW)	Write Data Valid to DS (Write) ↓ Delay	50		35		25		2,3
15	TdDS(DW)	DS ↑ to Write Data Not Valid Delay	60		35		30		2,3
16	TdA(DR)	Address Valid to Read Data Required Valid		410		255		200	1,2,3
17	TdAS(DS)	AS ↑ to DS ↓ Delay	80		55		40		2,3

NOTES:
1. When using extended memory timing add 2 TpC.
2. Timing numbers given are for minimum TpC.
3. See clock cycle time dependent characteristics table.
4. 16 MHz timing is preliminary and subject to change.

* All units in nanoseconds (ns).
† Test Load 1
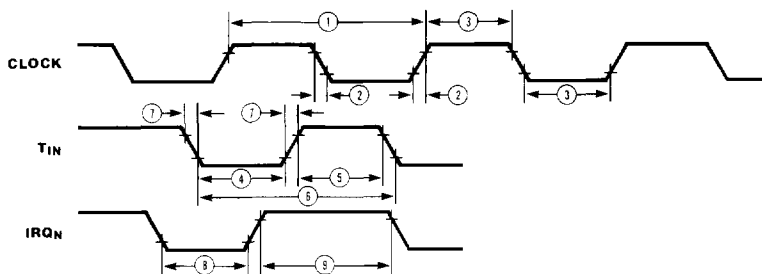° All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".

Figure 21. Additional Timing

AC CHARACTERISTICS
Additional Timing Table

Number	Symbol	Parameter	Z8681/82 8 MHz Min	Z8681/82 8 MHz Max	Z8681 12 MHz Min	Z8681 12 MHz Max	Z8681 16 MHz Min	Z8681 16 MHz Max	Notes
1	TpC	Input Clock Period	125	1000	83	1000	62.5	1000	1
2	TrC,TfC	Clock Input Rise and Fall Times		25		15		10	1
3	TwC	Input Clock Width	37		70		21		1
4	TwTinL	Timer Input Low Width	100		70		50		2
5	TwTinH	Timer Input High Width	3TpC		3TpC		3TpC		2
6	TpTin	Timer Input Period	8TpC		8TpC		8TpC		2
7	TrTin,TfTin	Timer Input Rise and Fall Times		100		100		100	2
8A	TwIL	Interrupt Request Input Low Time	100		70		50		2,4
8B	TwIL	Interrupt Request Input Low Time	3TpC		3TpC		3TpC		2,5
9	TwIH	Interrupt Request Input High Time	3TpC		3TpC		3TpC		2,3

NOTES:
1. Clock timing references use 3.8V for a logic "1" and 0.8V for a logic "0".
2. Timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
3. Interrupt request via Port 3.
4. Interrupt request via Port 3 ($P3_1$-$P3_3$)
5. Interrupt request via Port 3 ($P3_0$)
6. 16 MHz timing is preliminary and subject to change.
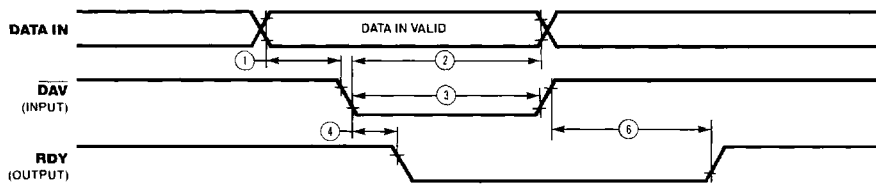* Units in nanoseconds (ns).
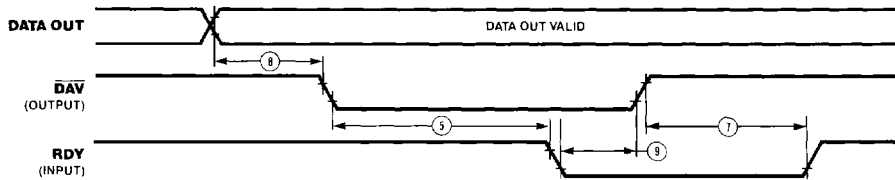
Figure 22a. Input Handshake Timing

Figure 22b. Output Handshake Timing

AC CHARACTERISTICS
Handshake Timing

Number	Symbol	Parameter	Z8681/82 8 MHz Min	Max	Z8681 12 MHz Min	Max	Z8681 16 MHz Min	Max	Notes
1	TsDI(DAV)	Data In Setup Time	0		0		0		
2	ThDI(DAV)	Data In Hold Time	230		160		145		
3	TwDAV	Data Available Width	175		120		110		
4	TdDAVIf(RDY)	DAV ↓ Input to RDY ↓ Delay		175		120		115	1,2
5	TdDAVOf(RDY)	DAV ↓ Output to RDY ↓ Delay	0		0		0		1,3
6	TdDAVIr(RDY)	DAV ↑ Input to RDY ↑ Delay		175		120		115	1,2
7	TdDAVOr(RDY)	DAV ↑ Output to RDY ↑ Delay	0		0		0		1,3
8	TdDO(DAV)	Data Out to DAV ↓ Delay	50		30		30		1
9	TdRDY(DAV)	Rdy ↓ Input to DAV ↑ Delay	0	200	0	140	0	130	1

NOTES:
1. Test load 1
2. Input handshake
3. Output handshake
4. 16 MHz timing is preliminary and subject to change.
† All timing references use 2.0V for a logic "1" and 0.8V for a logic "0".
* Units in nanoseconds (ns).

CLOCK CYCLE TIME-DEPENDENT CHARACTERISTICS

Number	Symbol	Z8681/82 8 MHz Equation	Z8681/82 12 MHz Equation
1	TdA(AS)	TpC-75	TpC-50
2	TdAS(A)	TpC-55	TpC-40
3	TdAS(DR)	4TpC-140 *	4TpC-110 *
4	TwAS	TpC-45	TpC-30
6	TwDSR	3TpC-125 *	3TpC-65 *
7	TwDSW	2TpC-90 *	2TpC-55 *
8	TdDSR(DR)	3TpC-175 *	3TpC-120 *
10	Td(DS)A	TpC-55	TpC-40
11	TdDS(AS)	TpC-55	TpC-30
12	TdR/W(AS)	TpC-75	TpC-55
13	TdDS(R/W)	TpC-65	TpC-50
14	TdDW(DSW)	TpC-75	TpC-50
15	TdDS(DW)	TpC-55	TpC-40
16	TdA(DR)	5TpC-215 *	5TpC-160 *
17	TdAS(DS)	TpC-45	TpC-30

* Add 2TpC when using extended memory timing