Z8 BASIC/Debug Interpreter

- The Z8671 MCU is a complete microcomputer preprogrammed with a BASIC/Debug interpreter. Interaction between the interpreter and its user is provided through an on-board UART.
- BASIC/Debug can directly address the Z8671's internal registers and all external memory. It provides quick examination and modification of any external memory location or I/O port.
- The BASIC/Debug interpreter can call machine language subroutines to increase execution speed.
- The Z8671's auto start-up capability allows a program to be executed on power-up or Reset without operator intervention.
- Single + 5 V power supply – all I/O pins TTL-compatible.
- Available in 8 MHz version.

General Description

The Z8671 Single-Chip Microcomputer (MCU) is one of a line of preprogrammed chips – in this case with a BASIC/Debug interpreter in ROM-offered by SGS. As a member of the Z8 Family of microcomputers, it offers the same abundance of resources as the other Z8 microcomputers.

Because the BASIC/Debug interpreter is already part of the chip circuit, programming is made much easier. The Z8671 MCU thus offers a combination of software and hardware that is ideal for many industrial applications. The Z8671 MCU allows fast hardware tests and bit-by-bit examination and modification of memory location, I/O ports, or registers. It also allows bit manipulation and logical operations. A self-contained line editor supports interactive debugging, further speeding up program development.

The BASIC/Debug interpreter, a subset of Dartmouth BASIC, operates with three kinds of memory: on-chip registers and external ROM or RAM. The BASIC/Debug interpreter is located in the 2K bytes of on-chip ROM.

Aditional features of the Z8671 MCU include the ability to call machine language subrotines to increase execution speed and the ability to have a program execute on power-up or Reset, without operator intervention.

Maximum memory addressing capabilities include 62K bytes of external program memory and 62K bytes of data memory with program storage beginning at location 800 hex. This provides up to 124K bytes of useable memory space. Very few 8-bit microcomputers can directly access this amount of memory.

Each Z8671 Microcomputer has 32 I/O lines, a 144-byte register file, an on-board UART, and two counter/timers.

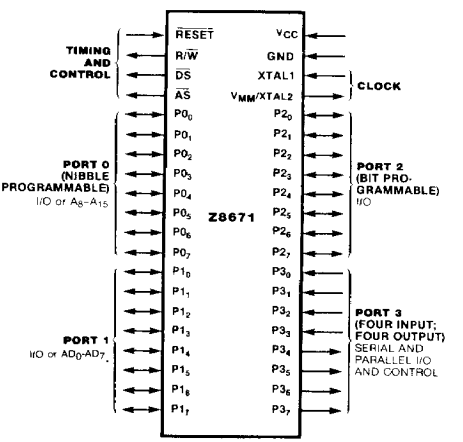


Figure 1. Logic Function

General Description (Continued)

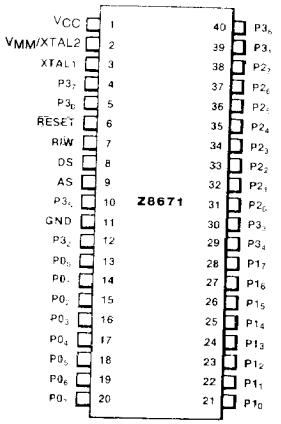


Figure 2a. Pin Configuration

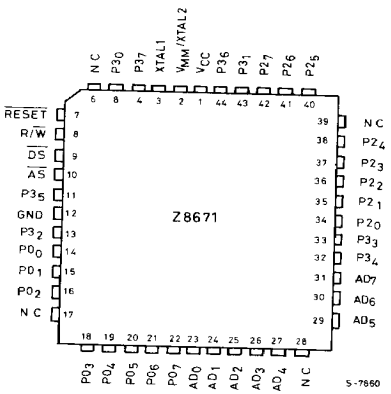


Figure 2b. Chip Carrier Pin Configuration

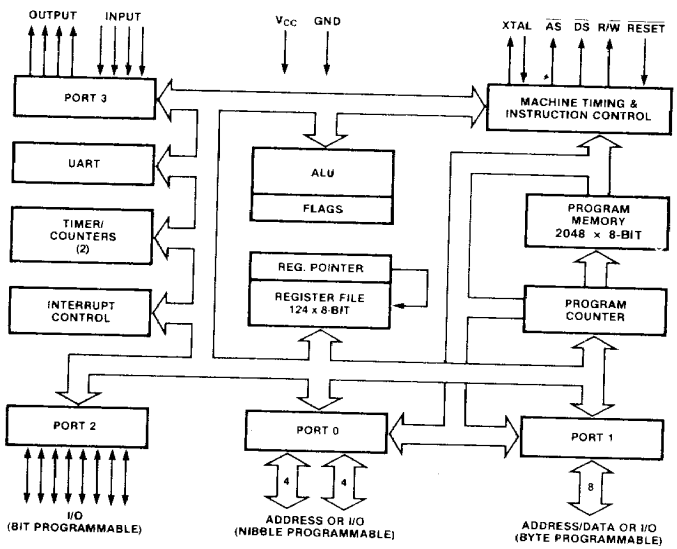


Figure 3. Functional Block Diagram

Architecture

Z8671 architecture is characterized by a flexible I/O scheme, and 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 Z8671 fulfills this with 32 pins dedicated to input and output. These lines are grouped into four 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/data bus for interfacing external memory.

Because the multiplexed address/data bus is merged with the I/O-oriented ports, the Z8671 can assume many different memory and I/O configurations. These configurations range from a self-contained microcomputer to a microprocessor that can address 124K bytes of external memory.

Three basic address spaces are available to support this wide range of configurations: program memory (internal and external), data memory (external) and the register file (internal). The 144-byte random-access register file is composed of 124 general-purpose registers, four 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.

Pin Description

$\overline{AS}$ *Address Strobe* (output, active Low). Address Strobe is pulsed once at the beginning of each machine cycle. Addressess output via Port 1 for all external program or data memory transfers are valid at the trailing edge of $\overline{AS}$. Under program control, $\overline{AS}$ can be placed in the high-impedance state along with Ports 0 and 1, Data Strobe and Read/Write.

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

$P0_0$-$P0_7$. *I/O Port Lines* (input/output, TTL compatible). 8 lines Nibble Programmable that can be configured under program control for I/O or external memory interface.

$P1_0$-$P1_7$. *I/O Port Lines* (input/output, TTL compatible). 8 lines Byte Programmable that can be configured under program control for I/O or multiplexed address (A_0-A_7) and data (D_0-D_7) lines used to interface with program/data memory.

$P2_0$-$P2_7$. *I/O Port Lines* (input/output, TTL compatible). 8 lines Bit Programmable. In addition they can be configured to provide open-drain outputs.

$P3_0$-$P3_4$. *Input Port Lines* (TTL compatible). They can also configured as control lines.

$P3_5$-$P3_7$. *Output Port Lines* (TTL compatible). They can also configured as control lines.

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ inizializes the Z8671. When $\overline{RESET}$ is deactivated, program execution begins from internal program location $000C_H$.

R/$\overline{W}$. *Read/Write* (output). R/$\overline{W}$ is Low when the Z8671 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 (8 or 12 MHz maximum) or an external single – phase clock (8 or 12 MHz maximum) to the on-chip clock oscillator and buffer.

Address Spaces

Program Memory. The Z8671's 16-bit program counter can address 64K bytes of program memory space. Program memory consists of 2K bytes of internal ROM and up to 62K bytes of external ROM, EPROM, or RAM. The first 12 bytes of program memory are reserved for interrupt vectors (Figure 4). These locations contain six 16-bit vectors that correspond to the six available interrupts. The BASIC/Debug interpreter is located in the 2K bytes of intenal ROM. The interpreter begins at address 12 and extends to 2047.

Data Memory. The Z8671 can address up to 62K bytes of external data memory beginning at location 2048 (Figure 5). External data memory may be included with, or separated from, the external program memory space. DM, an optional I/O function that can be programmed to appear on pin $P3_4$, is used to distinguish data and program memory space.

Register File. The 144-byte register file may be accessed by BASIC programs as memory locations 0 – 127 and 240 – 255. The register file includes four I/O port registers (R0-R3), 124 general-purpose registers (R4-R127), and 16 control and status registers (Figure 6).

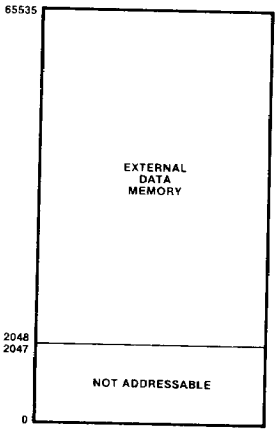


Figure 5. Data Memory Map

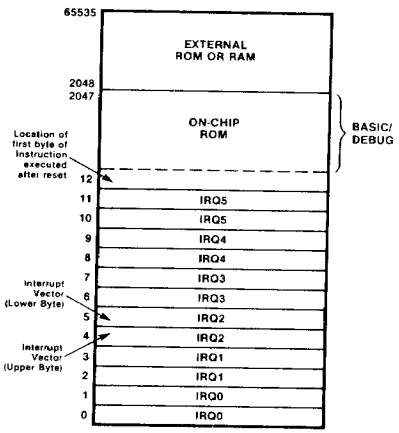


Figure 4. Programm Memory Map

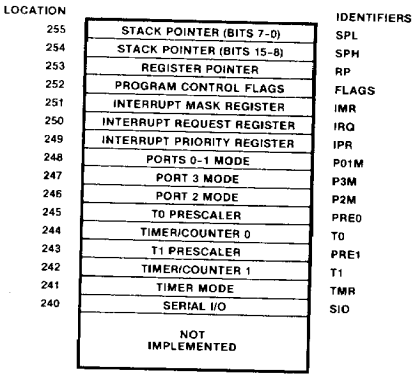


Figure 6. Control and Status Registers

Address Spaces (Continued)

The BASIC/Debug Interpreter uses many of the general-purpose registers as pointers, scratch workspace, and internal variables. Consequently, these registers cannot be used by a machine language subroutine or other user programs. On power-up/Reset, BASIC/Debug searches for external RAM memory and checks for an auto start-up program. In a non-destructive method, memory is tested at relative location xxFD(hex). When BASIC/Debug discovers RAM in the system, it initializes the pointer registers to mark the boundaries between areas of memory that are assigned specific uses. The top page of RAM is allocated for the line buffer, variable storage, and the GOSUB stack. Figure 7a illustrates the contents of the general-purpose registers in the Z8671 system with external RAM. When BASIC/Debug tests memory and finds no RAM, it uses an internal stack and shares register space with the input line buffer and variables. Figure 7b illustrates the contents of the general-purpose registers in the Z8671 system without external RAM.

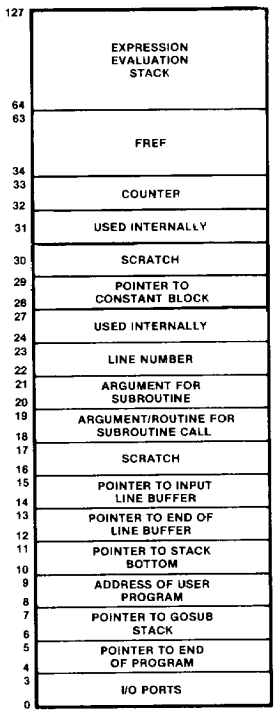


Figure 7a. General-Purpose Registers with External RAM

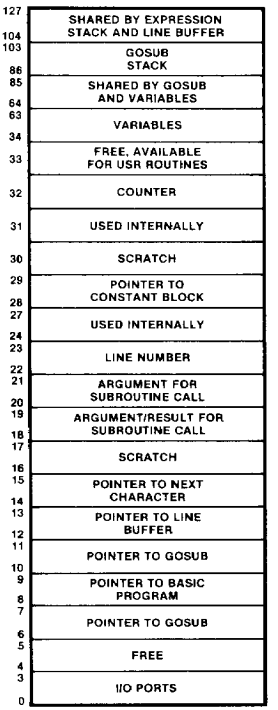


Figure 7b. General-Purpose Registers without External RAM

Address Spaces (Continued)

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 between location 2048 and 65535. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124 general-purpose registers (R4-R127).

Register Addressing. Z8671 instructions can access registers directly or indirectly with an 8-bit address field. The Z8671 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 group consisting of 16 continguous registers (Figure 8). The Register Pointer addresses the starting location of the active working-register group.

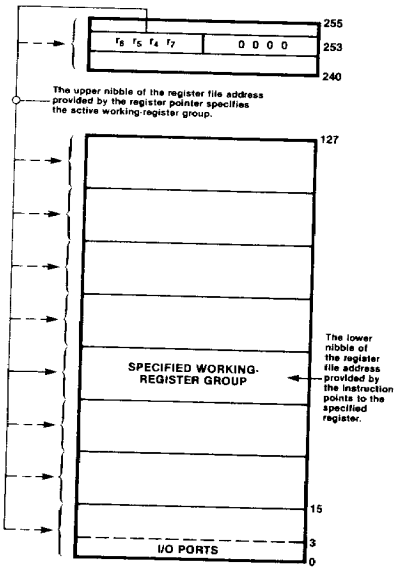


Figure 8. The Register Pointer

Program Execution

Automatic Start-up. The Z8671 has an automatic start-up capability which allows a program stored in ROM to be executed without operator intervention. Automatic execution occurs on power-on or Reset when the program is stored at address 1020 (hex).

Execution Modes. The Z8671's BASIC/Debug Interpreter operates in two execution modes: Run and Immediate. Programs are edited and interactively debugged in the Immediate mode. Some BASIC/Debug commands are used almost exclusively in this mode. The Run mode is entered from the Immediate mode by entering the command RUN. If there is a program in RAM, it is executed. The system returns to the Immediate mode when program execution is complete or interrupted by an error.

Interactive Debugging

Interactive debugging is accomplished with the self-contained line editor which operates in the Immediate mode. In addition to changing program lines, the editor can correct an immediate command before it is executed. It also allows the correction of typing and other errors as a program is entered.

BASIC/Debug allows interruptions and changes during a program run to correct errors and add new instructions without disturbing the sequential execution of the program. A program run is interrupted with the use of the escape key. The run is restarted with a GOTO command (followed by the appropriate line number) after the desired changes are entered. The same procedure is used to enter corrections after BASIC/Debug returns an error.

Commands

BASIC/Debug recognizes 15 command keywords. For detailed instructions of command usage, refer to the BASIC/Debug software manual.

GO	The GO command unconditionally branches to a machine language subrotine. This statement is similar to the USR function except that no value is returned by the assembly language routine.
GOSUB	GOSUB unconditionally branches to a subroutine at a line number specified by the user.
GOTO	GOTO unconditionally changes the sequence of program execution (branches to a line number).
IF/THEN	This commands is used for conditional operations and branches.
INPUT/IN	These commands request information from the user with the prompt "?", then read the input values (which must be separated by commas) from the keyboard, and store them in the indicated variables. INPUT discards any values remaining in the buffer from previous IN, INPUT, or RUN statements, and requests new data from the operator. IN uses any values left in the buffer first, then requests new data.
LET	LET assigns the value of an expression to a variable or memory location.
LIST	This command is used in the interactive mode to generate a listing of program lines stored in memory on the terminal device.
NEW	The NEW command resets pointer R10-11 to the beginning of user memory, thereby marking the space as empty and ready to store a new program.
PRINT	PRINT lists its arguments, which may be text messages or numerical values, on the output terminal.
REM	This command is used to insert explanatory messages into the program.
RETURN	This command returns control to the line following a GOSUB statement.
RUN	RUN initiates sequential execution of all instructions in the current program.
STOP	STOP ends program execution and clears the GOSUB stack.

Functions

BASIC/Debug supports two functions: AND and USR.

The AND function performs a logical AND. It can be used to mask, turn off, or isolate bits. This function is used in the following format:

AND (expression, expression)

The two expressions are evaluated, and their bit patterns are ANDed together. If only one value is included in the parentheses, it is ANDed with itself. A logical OR can also be performed by complementing the AND function. This is accomplished by subtracting each expression from −1. For example, the function below is equivalent to the OR of A and B.

−1−AND(−1−A, −1−B)

The USR function calls a machine language subroutine and returns a value. This is useful for applications in wich a subroutine can be performed more quickly and efficiently in machine language than in BASIC/Debug.

The address of the first instruction of the subroutine is the first argument of the USR function. The address can be followed by one or two values to be processed by the subroutine. In the following example, BASIC/Debug executes the subroutine located at address 2000 using values literal 256 and variable C.

USR(%2000,256,C)

The resulting value is stored in Registers 18-19.

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, with a maximum rate of 62.5K bit/second for 8MHz, and a maximum rate of 94.8K bit/second for 12MHz parts.

The Z8671 automatically adds a start bit and two stop bits to transmitted data (Figure 9). 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 (IRQ_4) 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 IRQ_3 interrupt request.

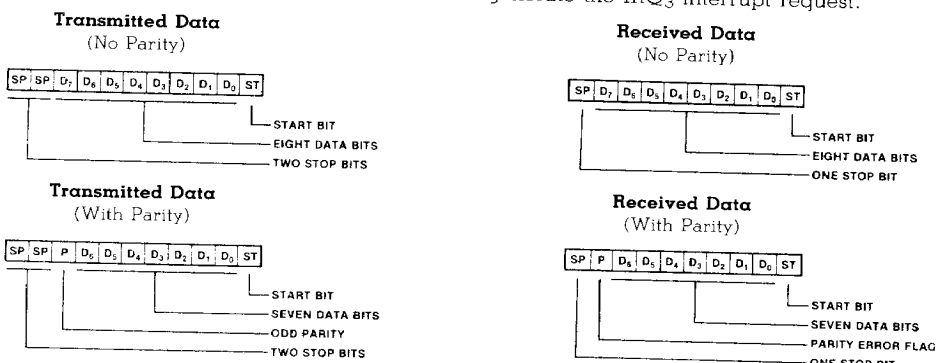


Figure 9. Serial Data Formats

I/O Ports

The Z8671 has 32 lines dedicated to input and output. These lines are grouped into four ports of eight lines each and are configurable as input, output or address/data. 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 can be programmed as a byte I/O port or as an address/data port for interfacing external memory. When used as an I/O port, Port 1 may be placed under handshake control. In this configuration, Port 3 lines $P3_3$ and $P3_4$ are used as the handshake controls RDY_1 and DAV_1 (Ready and Data Available).

Memory locations greater than 2048 are referenced through Port 1. To interface external memory, Port 1 must be programmed for the multiplexed Address/Data mode. If more than 256 external locations are required, Port 0 must output the additional lines.

Port 1 can be placed in the high-impedance state along with Port 0, $\overline{AS}$, $\overline{DS}$ and $R/\overline{W}$, allowing the Z8671 to share common resources in multiprocessor and DMA applications. Data transfers can be controlled by assigning $P3_3$ as a Bus Acknowledge input and $P3_4$ as a Bus Request output.

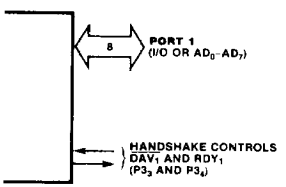


Figure 10a. Port 1

Port 0 can be programmed as a nibble I/O port, or as an address port for interfacing external memory. When used as an I/O port, Port 0 may be placed under handshake control. In this configuration, Port 3 lines $P3_2$ and $P3_5$ are used as the handshake controls $\overline{DAV}_0$ 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 nibble) 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. When Port 0 nibbles as defined as address bits, they can be set to the high-impedance state along with Port 1 and the control signals $\overline{AS}$, $\overline{DS}$ and $R/\overline{W}$.

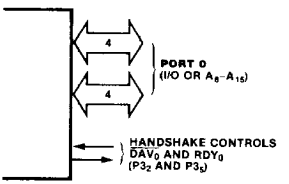


Figure 10b. Port 0

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

Like Ports 0 and 1, 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.

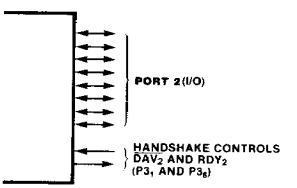


Figure 10c. Port 2

I/O Ports (Continued)

Port 3 lines can be configured as I/O or control lines. 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 Port 0, 1 and 2 ($\overline{DAV}$ and RDY); four external interrupt request signals (IRQ_0-IRQ_3); timer input and output signals (T_{IN} and T_{OUT}) and Data Memory Select ($\overline{DM}$).

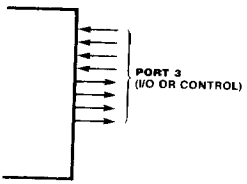


Figure 10d. Port 3

Counter/Timers

The Z8671 contains two 8-bit programmable counter/timers (T_0 and T_1), each driven by its own 6-bit programmable prescaler. The T_1 prescalers can be driven by internal or external clock sources; however, the T_0 prescaler is driven by the nternal 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 – IRQ_4 (T_0) or IRQ_5 (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 (4 MHz maximum for the 8 MHz device and 6 MHz maximum for the 12 MHz device) 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 non-retriggerable, 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 can be output.

Interrupts

The Z8671 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 Z8671 interrupts are vectored; however, the internal UART operates in a polling fashion. 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.

The BASIC/Debug Interpreter does not process interrupts. Interrupts are vectored through locations in internal ROM which point to addresses 1000-1011 (hex). To process interrupts, jump instructions can be entered to the interrupt handling routines at the appropriate addresses as shown in Table 1.

Address (hex)	Contains Jump Instruction and Sobroutine Address for:
1000-1002	IRQ_0
1003-1005	IRQ_1
1006-1008	IRQ_2
1009-100B	IRQ_3
100C-100E	IRQ_4
100F-1011	IRQ_5

Table 1. Interrupt Jump instruction

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 ($C_L \leq 15$ pF maximum) from each pin to ground. The specifications for the crystal are as follows:

- AT cut, parallel resonant
- Fundamental type, 8 MHz maximum
- Series resistance, $R_s \leq 100\ \Omega$

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 ← 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 arithmetical falgs plus two user selectable flags:

C	Z	S	V	D	H	F2	F1	

C	Carry falg		
Z	Zero flag		
S	Sign flag		
V	Overflow flag		
D	Decimal-adjust flag	**F1**	user flags
H	Half-carry flag	**F2**	user flags

Affected flags are indicated by:

0	Clared to zero
1	Set to one
*	Set or cleared according to operation
–	Unaffected
X	Undefined

Conditions 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

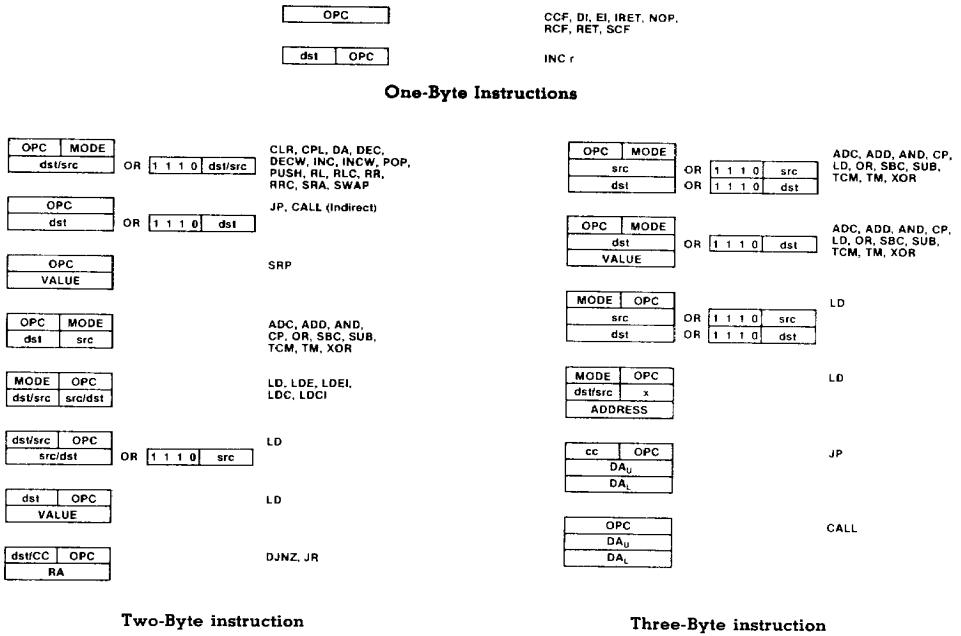


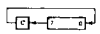
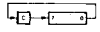
Figure 11. Instruction Formats

Instruction Summary

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected 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	–	–
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	–	*	*	*	–	–
IRET FLAGS ← @SP; SP ← SP + 1 PC ← @SP; SP ← SP + 2; IMR (7) ← 1			BF	*	*	*	*	*	*

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
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 r X r Ir R R IR IR	Im R r X r Ir r R IR Im R	rC r8 r9 r = 0-F 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 Summary (Continued)

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected 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 – –

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C Z S V D H
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 – –

Addr Mode dst	Addr Mode src	Lower Opcode Nibble
r	r	2
r	Ir	3
R	R	4
R	IR	5
R	IM	6
IR	IM	7

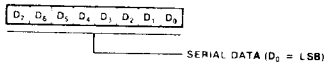
Note 1

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.

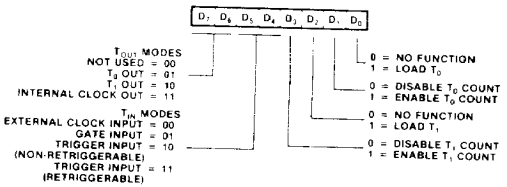
For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

Registers

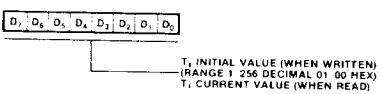
R240 SIO
Serial I/O Register
($F0_H$; Read/Write)



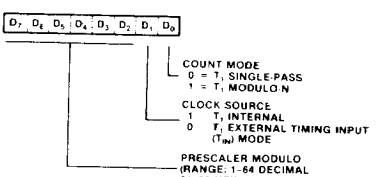
R241 TMR
Timer Mode Register
($F1_H$; Read/Write)



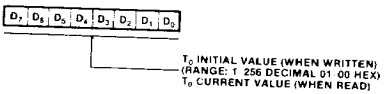
R242 T1
Counter Timer 1 Register
($F2_H$; Read/Write)



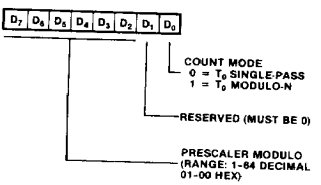
R243 PRE1
Prescaler 1 Register
($F3_H$; Write Only)



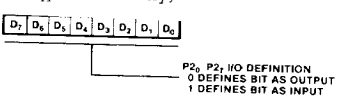
R244 T0
Counter/Timer 0 Register
($F4_H$; Read/Write)



R245 PRE0
Prescaler 0 Register
($F5_H$; Write Only)



R246 P2M
Port 2 Mode Register
($F6_H$; Write Only)



R247 P3M
Port 3 Mode Register
($F7_H$; Write Only)

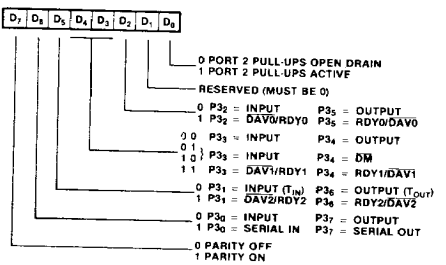
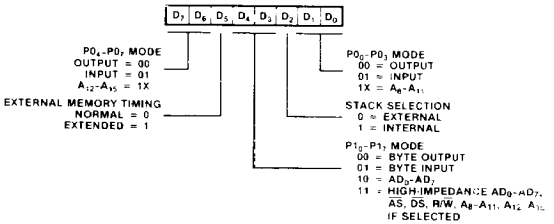


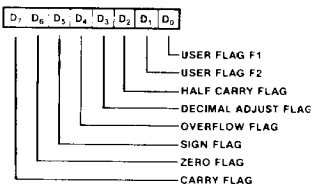
Figure 12. Control Registers

Registers (Continued)

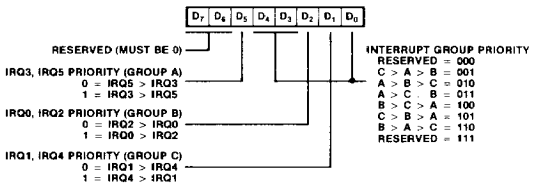
R248 P01M
Port 0 and 1 Mode Register
($F8_H$; Write Only)



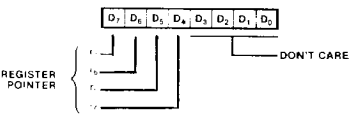
R252 FLAGS
Flag Register
(FC_H; Read/Write)



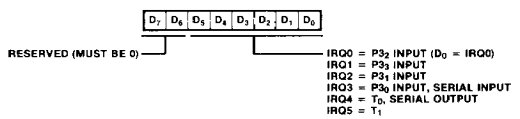
R249 IPR
Interrupt Priority Register
($F9_H$; Write Only)



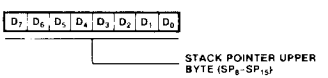
R253 RP
Register Pointer
(FD_H; Read/Write)



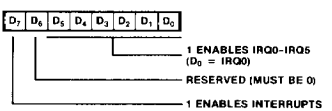
R250 IRQ
Interrupt Request Register
(FA_H; Read/Write)



R254 SHP
Stack Pointer
(FE_H; Read/Write)



R251 IMR
Interrupt Mask Register
(FB_H; Read/Write)



R255 SPL
Stack Pointer
(FF_H; Read/Write)

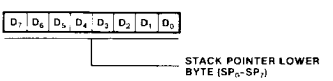
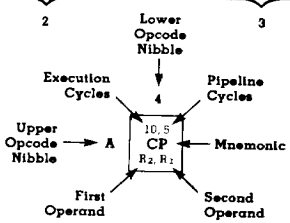


Figure 12. Control Registers (Continued)

Opcode Map

Lower Nibble (Hex)

Upper Nibble (Hex)	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 Ir1,r2		10,5 LD R2,IR1										6,0 NOP
Bytes per Instruction	2				3				2				3		1	



Legend:
R = 8-Bit Address
r = 4-Bit Address
R1 or r1 = Dst Address
R2 or r2 = 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

Voltage on all pins with respect to GND	−0.3 V to +7.0 V
Operating Ambient Temperature	0°C to +70°C
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 characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the reference pin. Standard conditions are as follows:

- +4.75 V ≤ V_{CC} ≤ +5.25 V
- GND = 0 V
- 0°C ≤ +70°C

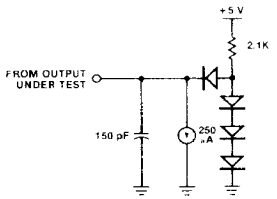


Figure 13. Test Load 1

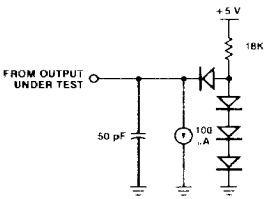


Figure 14. Test Load 2

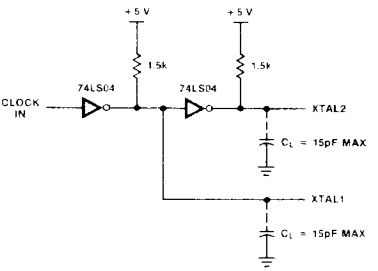


Figure 15. TTL External Clock Interface Circuit (Both the clock and its complement are required)

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	
V_{RL}	Reset Input Low Voltage	−0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −250 µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage	−10	10	µA	0 V ≤ V_{IN} ≤ +5.25 V
I_{OL}	Output Leakage	−10	10	µA	0 V ≤ V_{IN} ≤ +5.25 V
I_{IR}	Reset Input Current		−50	µA	V_{CC} = +5.25 V, V_{RL} = 0 V
I_{CC}	V_{CC} Supply Current		120	mA	
I_{MM}	V_{MM} Supply Current		10	mA	Power Down Mode
V_{MM}	Backup Supply Voltage	3	V_{CC}	V	Power Down

External I/O or Memory Read/Write

No.	Symbol	Parameter	Z8671 Min	Z8671 Max	Notes*†
1	TdA(AS)	Address Valid to $\overline{AS}$ ↑ Delay	50		1,2,3
2	TdAS(A)	$\overline{AS}$ ↑ to Address Float Delay	70		1,2,3
3	TdA(DS)	$\overline{AS}$ ↑ to Read Data Required Valid		360	1,2,3,4
4	TwAS	$\overline{AS}$ Low Width	80		1,2,3
5	TdAz(DS)	Address Float to $\overline{DS}$ ↓	0		1
6	TwDSR	$\overline{DS}$ (Read) Low Width	250		1,2,3,4
7	TwDSW	$\overline{DS}$ (Write) low Width	160		1,2,3,4
8	TdDSR(DR)	$\overline{DS}$ ↓ to Read Data Required Valid		200	1,2,3,4
9	ThDR(DS)	Read Data to $\overline{DS}$ ↑ Hold Time	0		1
10	TdDS(A)	$\overline{DS}$ ↑ to Address Active Delay	70		1,2,3
11	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		1,2,3
12	TdR/W(AS)	R/$\overline{W}$ Valid to $\overline{AS}$ ↑ Delay	50		1,2,3
13	TdDS(R/W)	$\overline{DS}$ ↑ to R/$\overline{W}$ Not Valid	60		1,2,3
14	TdDW(DSW)	Write Data Valid to $\overline{DS}$ (Write) ↓ Delay	50		1,2,3
15	TdDS(DW)	$\overline{DS}$ ↑ to Write data Not Valid Delay	70		1,2,3
16	TdA(DR)	Address valid to Read Data Required Valid		410	1,2,3,4
17	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	80		1,2,3

NOTES:
1. Test Load 1
2. Timing numbers given are for minimum TpC
3. Also see clock cycle time dependent characteristics table.
4. When using extended memory timing add 2 TpC.
5. All timing references use 2.0 V for a logic "1" and 0.8 for a logic "0".
* All units in nanoseconds (ns).
† All timings are preliminary and subject to change.

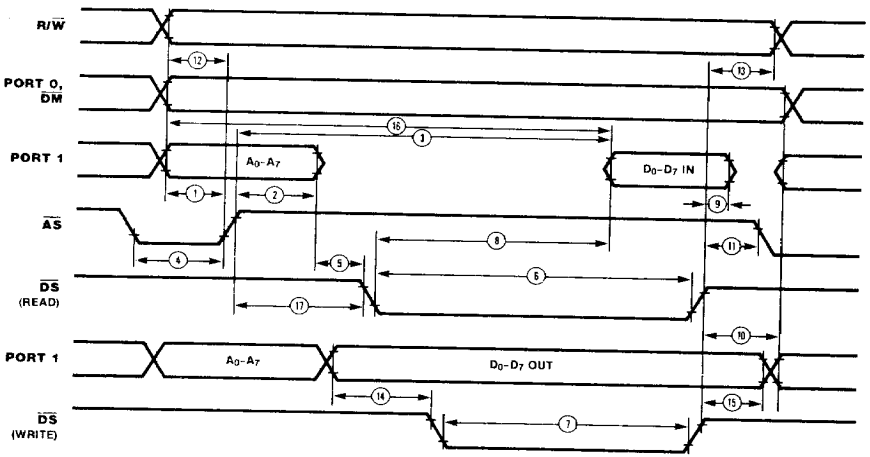


Figure 15. External I/O or Memory Read/Write

Additional Timing

No.	Symbol	Parameter	Z8671 Min	Z8671 Max	Notes*†
1	TpC	Input Clock Period	125	1000	1
2	TrC,TfC	Clock Input Rise And Fall Times		25	1
3	TwC	Input Clock Width	37		1
4	TwTinL	Timer Input low Width	100		2
5	TwTinH	Timer Input High Width	3TpC		2
6	TpTin	Timer Input Period	8TpC		2
7	TrTin,TfTin	Timer input Rise And Fall Times		100	2
8a	TwIL	Interrupt Request Input Low Time	100		2,3
8b	TwIL	Interrupt Request Input Low Time	3TpC		2,4
9	TwIH	Interrupt Request Input High Time	3TpC		2,3

NOTES:
1. Clock timing references uses 3.8 V for a logic "1" and 0.8 V for a logic "0"
2. Timing reference uses 2.0 V for a logic "1" and 0.8 V for a logic "0"
3. Interrupt request via Port 3 ($P3_1$-$P3_3$).
4. Interrupt request via Port 3 ($P3_0$).
* Units in nanoseconds (ns).
† All timings are preliminary and subject to change.

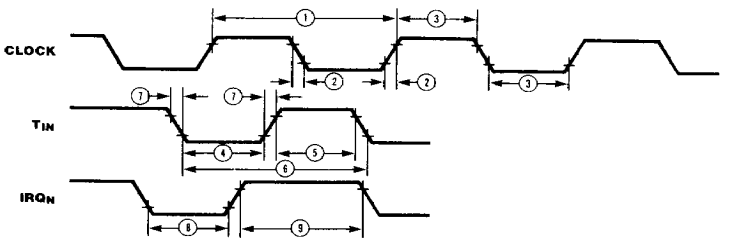


Figure 16. Additional Timing

Memory Port Timing

No.	Symbol	Parameter	Z8671 Min	Z8671 Max	Notes*†
1	TdA(DI)	Address Valid to Data Input Delay		460	1,2
2	ThDI(A)	Data In Hold Time	0		1

NOTES:
1. Test Load 2
2. This is a Clock-Cycle-Dependent parameter. For clock frequencies other than the maximum, use the following formula:
Z8671 = 5 TpC − 165; Z8671-12 = 5 TpC − 95

* Units are nanoseconds unless otherwise specified; timing are preliminary and subject to change.

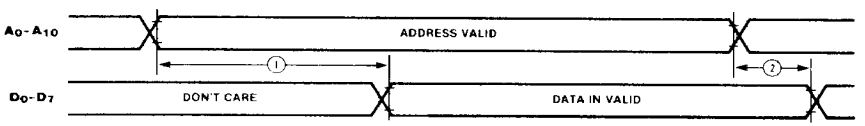


Figure 17. Memory Port Timing

Handshake Timing

No.	Symbol	Parameter	Z8671 Min	Z8671 Max	Notes*†
1	TsDI(DAV)	Data In Setup Time	0		
2	ThDI(DAV)	Data In Hold Time	230		
3	TwDAV	Data Available Width	175		
4	TdDAVI(RDY)	$\overline{DAV}$ ↓ Input to RDY ↓ Delay		175	1,2
5	TdDAVO(RDY)	$\overline{DAV}$ ↓ Output to RDY ↓ Delay	0		1,3
6	TdDAVIr(RDY)	$\overline{DAV}$ ↑ Input to RDY ↑ Delay		175	1,2
7	TdDAVOr(RDY)	$\overline{DAV}$ ↑ Output to RDY ↑ Delay	0		1,3
8	TdDO(DAV)	Data Out to $\overline{DAV}$ ↓ Delay	50		1
9	TdRDY(DAV)	Rdy ↓ Input to $\overline{DAV}$ ↑ Delay	0	200	1

NOTES:
1. Test load 1
2. Input handshake
3. Output handshake
4. All timing references use 2.0 V for a logic "1" and 0.8 V for a logic "0".

* Units in nanoseconds (ns).
† All timings are preliminary and subject to change.

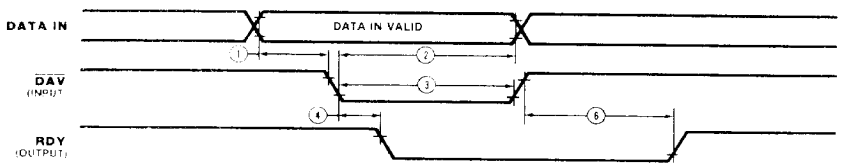


Figure 17a. Input Handshake

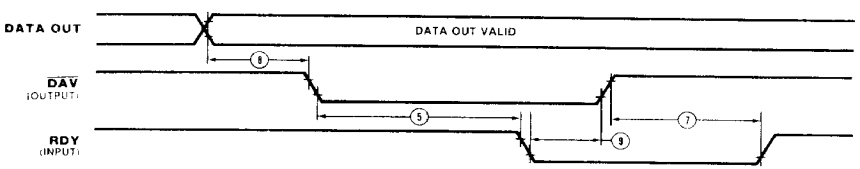


Figure 17b. Output Handshake

Clock-Cycle-Time-Dependent Characteristics

Number	Symbols	Z8671 Equation
1	TdA(AS)	TpC-75
2	TdAS(A)	TpC-55
3	TdAS(DR)	4TpC-140*
4	TwAS	TpC-45
6	TwDSR	3TpC-125*
7	TwDSW	2TpC-90*
8	TdDSR(DR)	3TpC-175*
10	Td(DS)A	TpC-55
11	TdDS(AS)	TpC-55
12	TdR/W(AS)	TpC-75
13	TdDS(R/W)	TpC-65
14	TdDW(DSW)	TpC-75
15	TdDS(DW)	TpC-55
16	TdA(DR)	5TpC-215*
17	TdAS(DS)	TpC-45

* Add 2TpC when using extended memory timing.

Ordering Information

Type	Package	Temp.	Clock	Description
Z8671 B1	Plastic	0/+70°C	8 MHz	Z8 Mask programmed with BASIC/Debug Interpreter
Z8671 B6	Plastic	−40/+85°C		
Z8671 D1	Ceramic	0/+70°C		
Z8671 D6	Ceramic	−40/+85°C		
Z8671 C1	Plastic Chip Carrier	0/+70°C		
Z8671 C6	Plastic Chip Carrier	−40/+85°C		
Z8671 K1	Ceramic Chip Carrier	0/+70°C		
Z8671 K6	Ceramic Chip Carrier	−40/+85°C		