ZiLOG

PRELIMINARY PRODUCT SPECIFICATION

Z88C00
CMOS SUPER8® ROMLESS MCU

FEATURES

- Full Super8 Instruction Set
- 1.2 micron CMOS technology
- Available in 44-, 48-, and 68-pin packages
- Multiply and Divide instructions, Boolean and BCD operations
- 325 byte registers, including 272 general-purpose registers, and 53 mode and control registers
- Addresses up to 128 Kbytes of external program and data memory
- Two register pointers allow 600 nsec access time
- Direct Memory Access (DMA)
- Two 16-bit counter/timers with 8-bit prescalers
- Up to 32 bit-programmable and byte-programmable I/O lines, with two handshake channels
- Interrupt structure supports:
  - 27 interrupt sources
  - 16 interrupt vectors
  - 8 interrupt levels
  - Servicing capabilities in 600 nsec
- Full-duplex UART with special features
- On-chip oscillator
- DC to 25 MHz operating frequency
- Two modes: STOP and HALT
- Low power consumption (<250 mW)
- Full pin-for-pin compatibility with NMOS Super8
- Watch-Dog Timer
- Optional Demultiplexed Z-BUS®

GENERAL DESCRIPTION

The CMOS Super8® offers new flexibility and sophistication in 8-bit microcontrollers. The Super8 offers all the features necessary for industrial, consumer, and automotive applications with an enhanced feature set in CMOS technology. At the same time, the CMOS Super8 retains full pin-for-pin compatibility with the NMOS Super8. Available in 48-pin DIP, and 44-, 68-pin PLCC, the CMOS Super8 is the last word in general purpose controllers.

The Super8 features a full-duplex, Universal Asynchronous Receiver/Transmitter (UART) with on-chip baud rate generator, on-chip oscillator, two 16-bit counter/timers each with an 8-bit prescaler, a Direct Memory Access controller (DMA), Watch-Dog Timer (WDT), STOP and HALT modes.

Incorporated in the new CMOS Super8 is an option for a de-multiplexed external memory interface bus. In de-multiplexed mode, Ports 0 and 4 function as address ports, with PORT 1 as data bus. PORT 4 drives the lower address bits and PORT 0 drives the upper address bits, when both ports are configured as external memory interface. This gives the user more addressing flexibility. This option is externally controlled by the de-mux pin on the 68-pin PLCC package, it is a bonding option on the 48-pin DIP package and is not supported on the 44-pin PLCC.

Finally, by adding the enhanced features of WDT, STOP and HALT modes, and more versatile counter/timers, the CMOS Super8 can fit easily into more complex function applications where a general-purpose microcontroller is a necessity.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

GENERAL DESCRIPTION (Continued)

Figure 1 is the functional block diagram of Z88C00. The device is housed in a 48-pin DIP (Figure 2), a 68-pin PLCC package (Figure 3) and a 44-pin PLCC package (Figure 4). The pin functions of the Z88C00 are shown in Figure 5.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

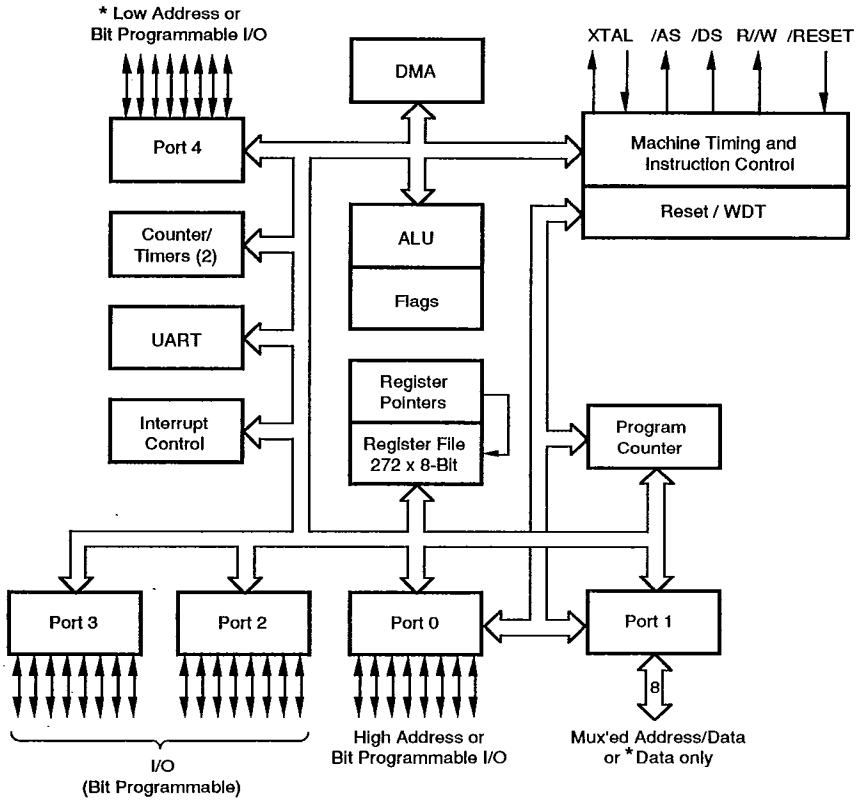


Figure 1. Functional Block Diagram

Powered by ICminer.com Electronic-Library Service CopyRight 2003

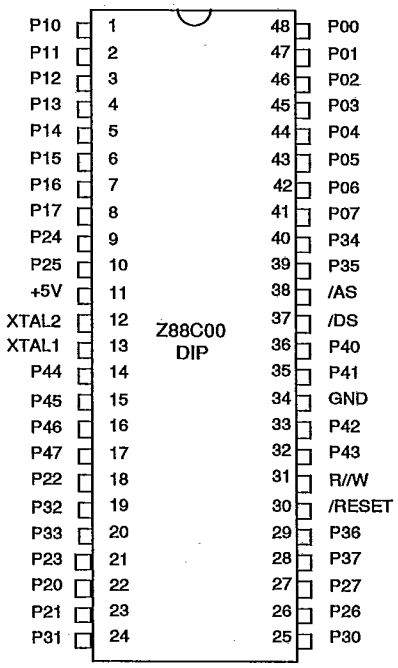


Figure 2. 48-Lead DIP Package

Table 1. 48-Lead DIP Pin Assignments

Pin #	Symbol	Function	Direction
1-8	P10-17	Port 1, pins 0,1,2,3,4,5,6,7	In/Output
9-10	P24-25	Port 2, pins 4,5	In/Output
11	V_{cc}	Power Supply	Input
12	XTAL2	Crystal Oscillator	Output
13	XTAL1	Crystal Oscillator	Input
14-17	P44-47	Port 4, pins 4,5,6,7	In/Output
18	P22	Port 2, pin 2	In/Output
19-20	P32-33	Port 3, pins 2,3	In/Output
21-23	P23-21	Port 2, pins 3,0,1	In/Output
24-25	P31-30	Port 3, pins 1,0	In/Output
26-27	P26-27	Port 2, pins 6,7	In/Output
28-29	P37-36	Port 3, pins 7,6	In/Output
30	/RESET	RESET	Input
31	R/W	READ/WRITE	Output
32-33	P43-42	Port 4, pins 3,2	In/Output
34	GND	Ground	Input
35-36	P41-40	Port 4, pins 1,0	In/Output
37	/DS	Data Strobe	Output
38	/AS	Address Strobe	Output
39-40	P35-34	Port 3, pins 5,4	In/Output
41-48	P07-00	Port 0, pins 7,6,5,4,3,2,1,0	In/Output

Powered by ICminer.com Electronic-Library Service CopyRight 2003

GENERAL DESCRIPTION (Continued)

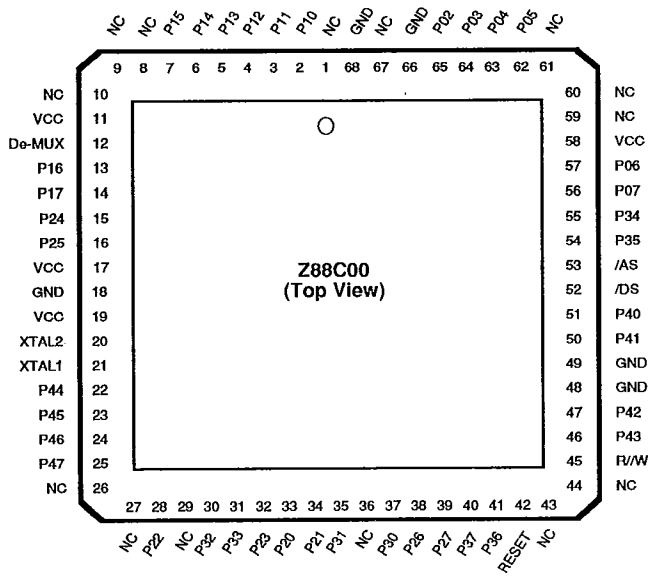


Figure 3. 68-Lead PLCC Package

Table 2. 68-Lead PLCC Pin Assignments

Pin #	Symbol	Function	Direction
1	NC	Not Connected	
2-7	P10-15	Port 1, pins 0,1,2,3,4,5	In/Output
8-10	NC	Not Connected	
11	V_{cc}	Power Supply	Input
12	De-Mux	De-multiplex Pin	Input
13-14	P16-17	Port 1, pins 6,7	In/Output
15-16	P24-25	Port 2, pins 4,5	In/Output
17	V_{cc}	Power Supply	Input
18	GND	Ground	Input
19	V_{cc}	Power Supply	Input
20	XTAL2	Crystal Oscillator	Output
21	XTAL1	Crystal Oscillator	Input
22-25	P44-47	Port 4, pins 4,5,6,7	In/Output
26-27	NC	Not Connected	
28	P22	Port 2, pin 2	In/Output
29	NC	Not Connected	
30-31	P32-33	Port 3, pins 2,3	In/Output
32-34	P23-21	Port 2, pins 3,0,1	In/Output
35	P31	Port 3, pin 1	In/Output
36	NC	Not Connected	

Pin #	Symbol	Function	Direction
37	P30	Port 3, pin 0	In/Output
38-39	P26-27	Port 2, pins 6,7	In/Output
40-41	P37-36	Port 3, pins 7,6	In/Output
42	/RESET	RESET	Input
43-44	NC	Not Connected	
45	R//W	READ/WRITE	Output
46-47	P43-42	Port 4, pins 3,2	In/Output
48-49	GND	Ground	Input
50-51	P41-40	Port 4, pins 1,0	In/Output
52	/DS	Data Strobe	Output
53	/AS	Address Strobe	Output
54-55	P43-42	Port 4, pins 3,2	In/Output
56-57	P07-06	Port 0, pins 7,6	In/Output
58	V_{cc}	Power Supply	Input
59-61	NC	Not Connected	
62-65	P05-02	Port 0, pins 5,4,3,2	In/Output
66	GND	Ground	Input
67	NC	Not Connected	
68	GND	Ground	Input

Powered by ICminer.com Electronic-Library Service CopyRight 2003

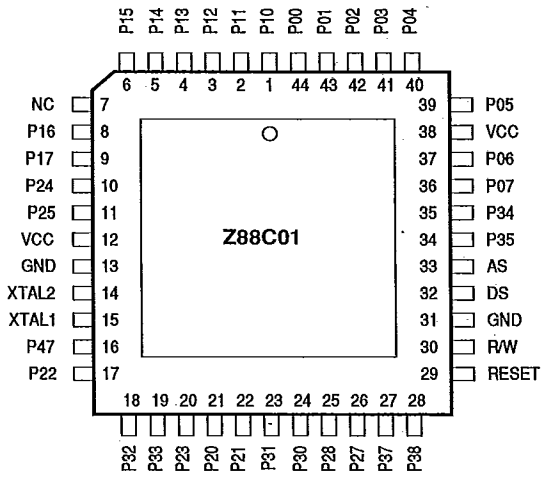


Figure 4. 44-Lead PLCC Package

Table 3. 44-Lead PLCC Pin Identification

Pin #	Symbol	Function	Direction	Pin #	Symbol	Function	Direction
1-6	P10-15	Port 1, pins 0,1,2,3,4,5	In/Output	23-24	P31-30	Port 3, pins 1,0	In/Output
7	NC	Not Connected		25-26	P26-27	Port 2, pins 6,7	In/Output
8-9	P16-17	Port 1, pins 6,7	In/Output	27-28	P37-36	Port 3, pins 7,6	In/Output
10-11	P24-25	Port 2, pins 4,5	In/Output	29	/RESET	Reset	Input
12	V_{cc}	Power Supply	Input	30	R/W	Read/Write	Output
13	GND	Ground	Input	31	GND	Ground	Input
14	XTAL2	Crystal Oscillator	Output	32	/DS	Data Strobe	Output
15	XTAL1	Crystal Oscillator	Input	33	/AS	Address Strobe	Output
16	P47	Port 4, pin 7	In/Output	34-35	P35-34	Port 3, pins 5,4	In/Output
17	P22	Port 2, pin 2	In/Output	36-37	P07-06	Port 0, pins 7,6	In/Output
18-19	P32-33	Port 3, pins 2,3	In/Output	38	V_{cc}	Power Supply	Input
20-22	P23-21	Port 2, pins 3,0,1	In/Output	39-44	P05-00	Port 0, pins 5,4,3,2,1,0	In/Output

Powered by ICminer.com Electronic-Library Service CopyRight 2003

GENERAL DESCRIPTION (Continued)

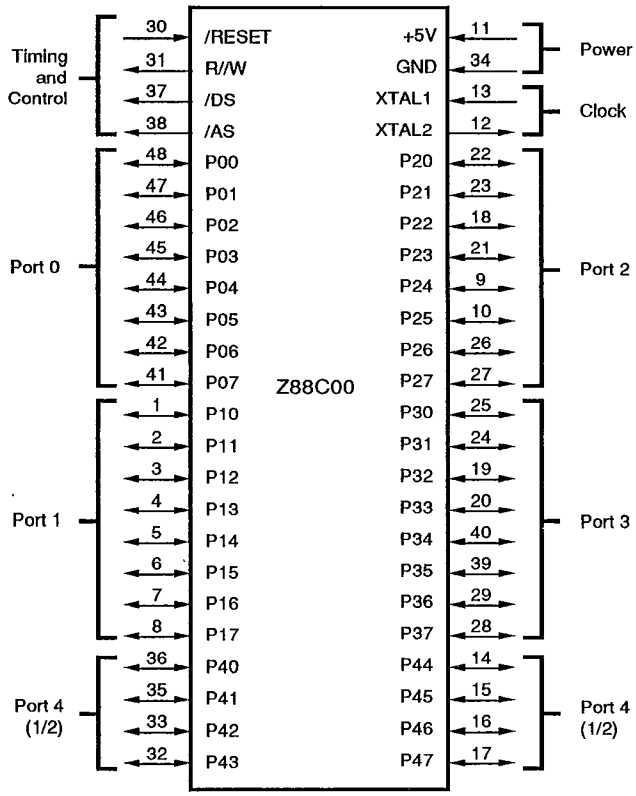


Figure 5. Pin Functions

Powered by ICminer.com Electronic-Library Service CopyRight 2003

ARCHITECTURE

The Super8 architecture includes 325 byte wide internal registers. 268 of these are available for general purpose use; the remaining 57 provide control and mode functions.

The instruction set is specially designed to deal with this large register set. It includes a full complement of 8-bit arithmetic and logical operations, including multiply and divide instructions and provisions for BCD operations. Addresses and counters can be incremented and decremented as 16-bit quantities. Rotate, shift, and bit manipulation instructions are provided. Three new instructions support threaded-code languages.

The UART is a full-function multipurpose asynchronous serial channel with many premium features.

The 16-bit counters can operate independently or be cascaded to perform 32-bit counting and timing operations. The DMA controller handles transfers to and from the register file or memory. DMA can use the UART or one of two ports with handshake capability.

All input pins on the Super8 will be provided with weak latches. Weak latches on inputs prevent them from floating which is highly desirable in STOP or HALT mode to reduce standby current as well as in normal operating mode to reduce unnecessary current flow and to eliminate noise on a floating input.

Weak latches on inputs are automatically disabled when the corresponding output is configured as open drain.

The architecture appears in the block diagram (Figure 1).

PIN DESCRIPTIONS

The Super8 connects to external devices via the following TTL-compatible pins:

/AS Address Strobe *(output, active Low).* AS is pulsed Low once the beginning of each machine cycle. The rising edge indicates that addresses R/W and DM, when used are valid.

/DS Data Strobe *(output, active Low).* DS provides timing for data movement between the address/data bus and external memory. During write cycles, data outputs is valid at the leading edge of DS. During read cycles, data input must be valid prior to the trailing edge of DS.

P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, Port I/O Lines *(input/output).* These 40 lines are divided into five 8-bit I/O ports that can be configured under program control for I/O or external memory interface.

Port 1 is dedicated as a multiplexed address/data port or as a data bus alone in de-multiplexed mode, and Port 0 pins can be assigned as additional address lines; Port 0 non-address pins may be assigned as I/O.

Ports 2 and 3 can be assigned on a bit-for-bit basis as general I/O or interrupt lines. They can also be used as special-purpose I/O lines to support the UART, counter/timers, or handshake channels.

Port 4 is used for general I/O or as the lower address byte in de-mux mode.

During RESET, all port pins are configured as inputs (high impedance) except for Port 1 and Port 0. Port 1 is configured as a multiplexed address/data bus, and Port 0 pins P00-P04 are configured as address out, while pins P05-P07 are configured as inputs.

/RESET *(input, active Low).* Reset initializes and starts the Super8. When it is activated, it halts all processing; when it is deactivated, the Super8 begins processing at address 0020H.

R/W Read/Write *(output).* R/W determines the direction of data transfer for external memory transactions. It is Low when writing to program memory or data memory, and High for everything else.

XTAL1, XTAL2 *(Crystal oscillator input).* These pins connect a parallel resonant crystal or an external clock source to the on-board clock oscillator and buffer.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

REGISTERS

The Super8 contains a 256-byte internal register space. However, by using the upper 64 bytes of the register space more than once, a total of 325 registers are available.

Registers from 00 to BF are used only once. They can be accessed by any register command. Register addresses CO to FF contain two separate sets of 64 registers. One set, called control registers, can only be accessed by register direct commands. The other can only be addressed by register indirect, indexed, stack, and DMA commands.

The uppermost 32 register direct registers (E0 to FF) are further divided into two bands (0 and 1), selected by Bank Select bit in the Flag register. When a Register Direct command accesses a register between E0 and FF, it looks at the Bank Select bit in the Flag register to select one of the banks.

The register space is shown in Figure 6.

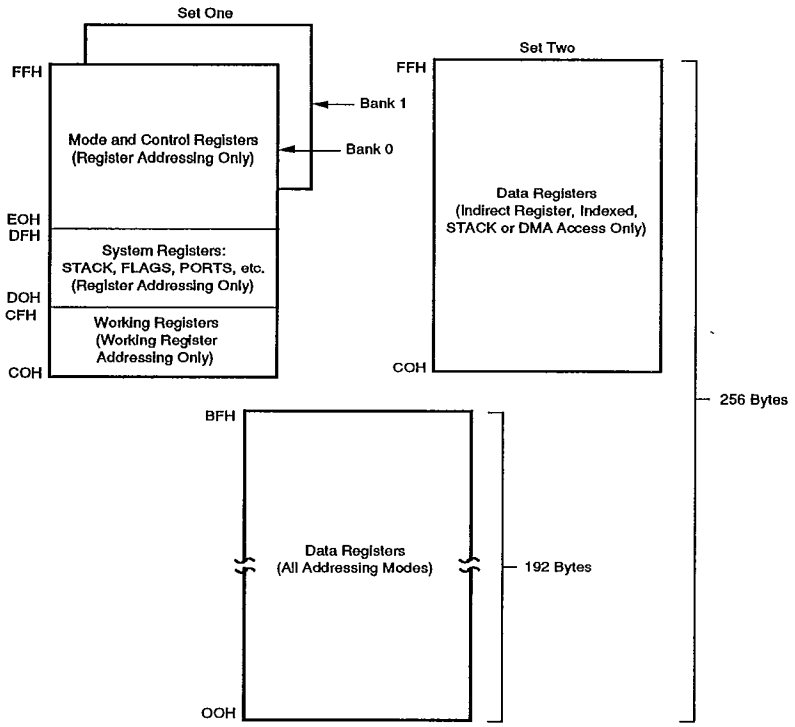


Figure 6. Super8 Registers

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Working Register Window

Control registers R214 and R215 are the register pointers RP0 and RP1. They each define a moveable, 8-register section of the register space. The registers within these spaces are called working registers.

Working registers can be accessed using short 4-bit addresses. The process, shown in a section of Figure 7, works as follows:

- The high order bit of the 4-bit address selects one of the two register pointers (0 selects RP0; 1 selects RP1).
- The five high order bits in the register pointer select an 8 register (contiguous) slice of the register space.
- The three low order bits of the 4-bit address selects one of the eight registers in the slice.

The net effect is to concatenate the five bits from the register pointer to the three bits from the address to form an 8-bit address. As long as the address in the register pointer remains unchanged, the three bits from the address will always point to an address within the same eight registers.

The register pointers can be moved by changing the five high bits in control registers R214 for RP9 and R215 for RP1.

The working registers can also be accessed by using full 8-bit addressing. When an 8-bit logical address in the range 192 to 207 (C0 to CF) is specified, the lower nibble is used similarly to the 4-bit addressing described above. This is shown in section b. of Figure 7.

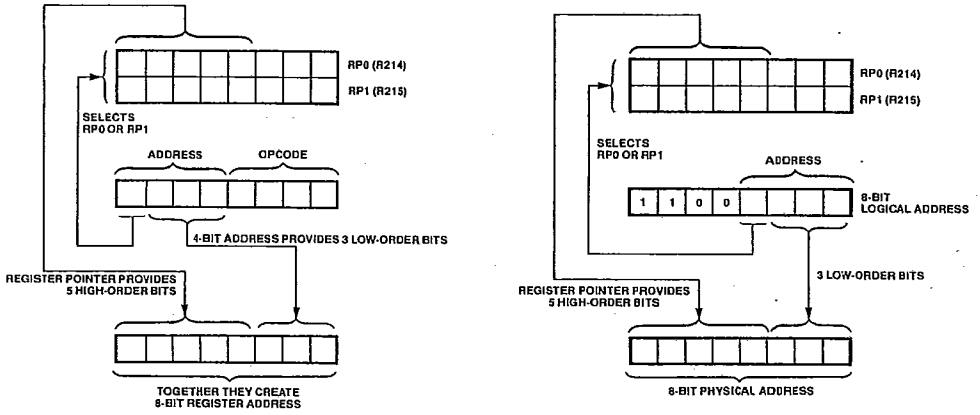


Figure 7. Working Register Window

Powered by ICminer.com Electronic-Library Service CopyRight 2003

REGISTERS (Continued)

Since any direct access to logical addresses 192 to 207 involves the register pointers, the physical registers 192 to 207 can only be accessed when selected by a register pointer. After a reset, RP0 points to R192 and RP1 points to R200.

Register List

Table 4 lists the Super8 registers. For more details see Figure 8.

Table 4. Super8 Registers

Address Decimal	Hexadecimal		Mnemonic	Function
General Purpose Registers				
000-192	00-BF		-	General purpose (all address modes)
192-207	C0-CF		-	Working register (direct only)
192-255	C0-FF		-	General purpose (indirect only)
Mode and Control Registers				
208	D0		P0	Port 0 I/O bits
209	D1		P1	Port 1 (I/O only)
210	D2		P2	Port 2
211	D3		P3	Port 3
212	D4		P4	Port 4
213	D5		FLAGS	System Flags Register
214	D6		RP0	Register Pointer 0
215	D7		RP1	Register Pointer 1
216	D8		SPH	Stack Pointer Low Byte
217	D9		SPL	Stack Pointer High Byte
218	DA		IPH	Instruction Pointer High Byte
219	DB		IPL	Instruction Pointer Low Byte
220	DC		IRQ	Interrupt Request
221	DD		IMR	Interrupt Mask Register
222	DE		SYM	System Mode
223	DF		HMR	Halt Mode Register
224	E0	Bank 0	C0CT	CTR 0 Control
		Bank 1	COM	CTR 0 Mode
225	E1	Bank 0	C1CT	CTR 1 Control
		Bank 1	C1M	CTR 1 Mode
226	E2	Bank 0	C0CH	CTR 0 Capture Register, bits 8-15
		Bank 1	C0TCH	CTR 0 Timer Constant, bits 8-15
227	E3	Bank 0	C0CL	CTR 0 Capture Register, bits 0-7
		Bank 1	C0TCL	CTR 0 Time Constant, bits 0-7
228	E4	Bank 0	C1CH	CTR 1 Capture Register, bits 8-15
		Bank 1	C1TCH	CTR 1 Time Constant, bits 8-15
229	E5	Bank 0	C1CL	CTR 1 Capture Register, bits 0-7
		Bank 1	C1TCL	CTR 1 Time Constant, bits 0-7
230	E6	Bank 0	CTPRS	Counter Prescaler
230	E6	Bank 1	WDTSMR	Watch-Dog/Stop Mode Register
235	EB	Bank 0	UTC	UART Transmit Control
236	EC	Bank 0	URC	UART Receive Control

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Address Decimal	Hexadecimal		Mnemonic	Function
Mode and Control Registers (Continued)				
237	ED	Bank 0	UIE	UART Interrupt Enable
238	EE	Bank 0	UTI	Transmit Interrupt Register
239	EF	Bank 0	UIO	UART Data
240	F0	Bank 0	P0M	Port 0 Mode
		Bank 1	DCH	DMA Count, bits 8-15
241	F1	Bank 0	PM	Port Mode Register
		Bank 1	DCL	DMA Count, bits 0-7
244	F4	Bank 0	H0C	Handshake Channel 0 Control
245	F5	Bank 0	H1C	Handshake Channel 1 Control
246	F6	Bank 0	P4D	Port 4 Direction
247	F7	Bank 0	P4OD	Port 4 Open Drain
248	F8	Bank 0	P2AM	Port 2/3 A Mode
		Bank 1	UBGH	UART Baud Rate Generator, bits 8-15
249	F9	Bank 0	P2BM	Port 2/3 B Mode
		Bank 1	UBGL	UART Baud Rate Generator, bits 0-7
250	FA	Bank 0	P2CM	Port 2/3 C Mode
		Bank 1	UMA	UART Mode A
251	FB	Bank 0	P2DM	Port 2/3 D Mode
		Bank 1	UMB	UART Mode B
252	FC	Bank 0	P2AIP	Port 2/3 A Interrupt Pending
253	FD	Bank 0	P2BIP	Port 2/3 B Interrupt Pending
254	FE	Bank 0	EMT	External Memory Timing
		Bank 1	WUMCH	Wake-up Match Register
255	FF	Bank 0	IPR	Interrupt Priority Register
		Bank 1	WUMSK	Wake-up Mask Register

Powered by ICminer.com Electronic-Library Service CopyRight 2003

MODE AND CONTROL REGISTERS

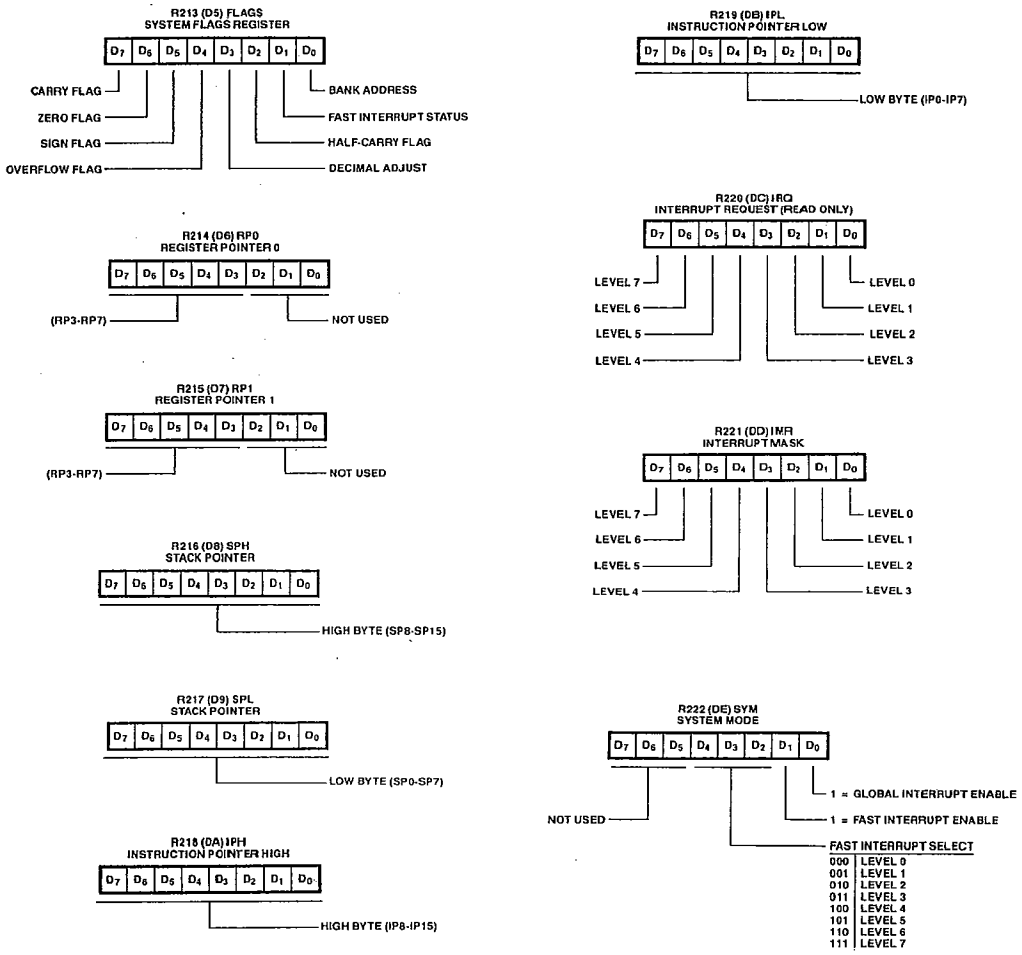


Figure 8. Mode and Control Registers

Powered by ICminer.com Electronic-Library Service CopyRight 2003

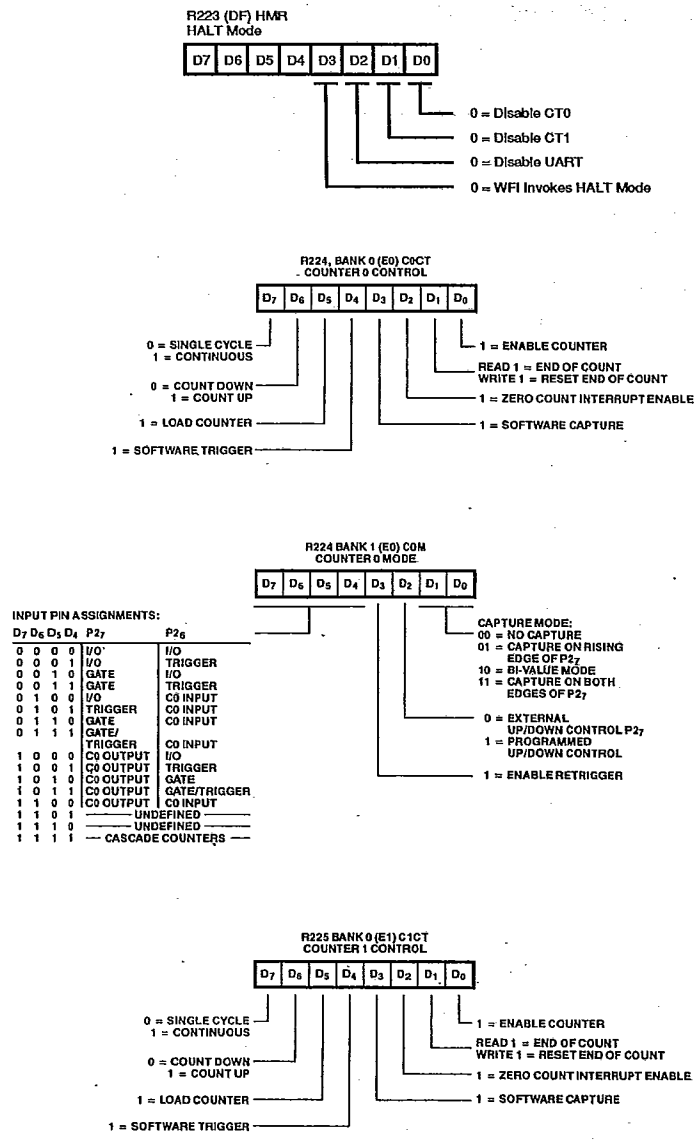


Figure 8. Mode and Control Registers (Continued)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

MODE AND CONTROL REGISTERS (Continued)

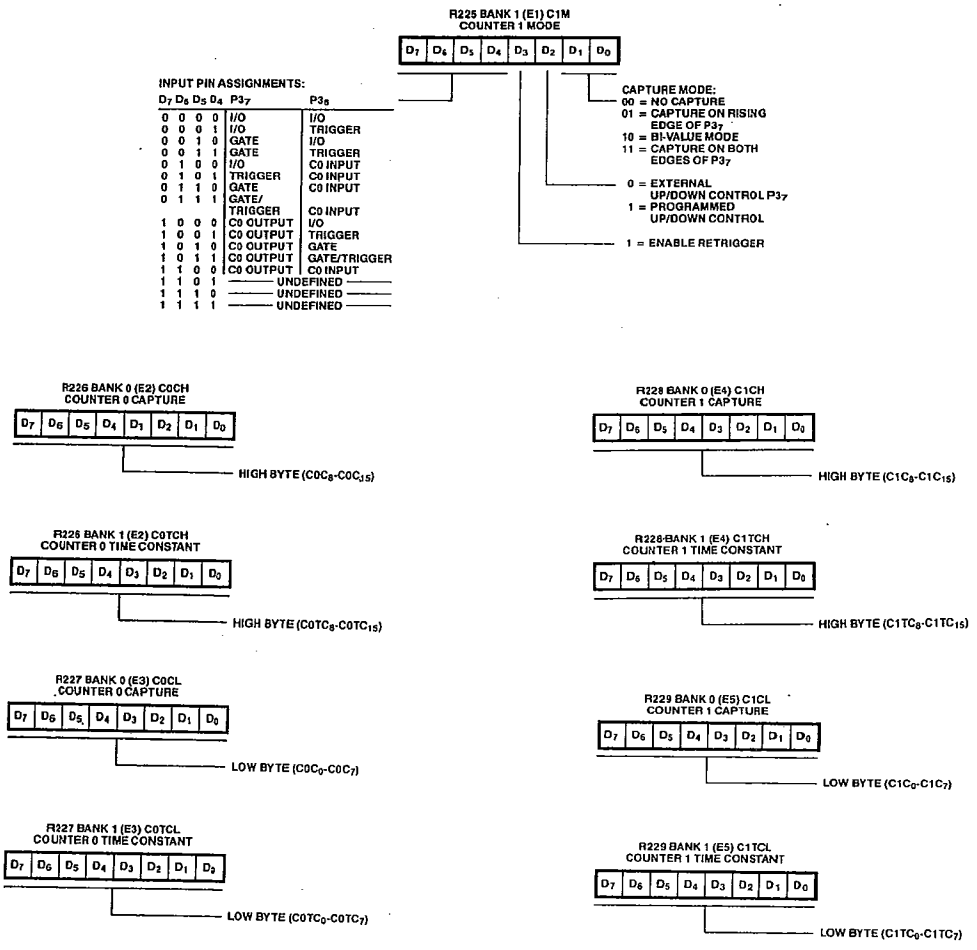


Figure 8. Mode and Control Registers

Powered by ICminer.com Electronic-Library Service CopyRight 2003

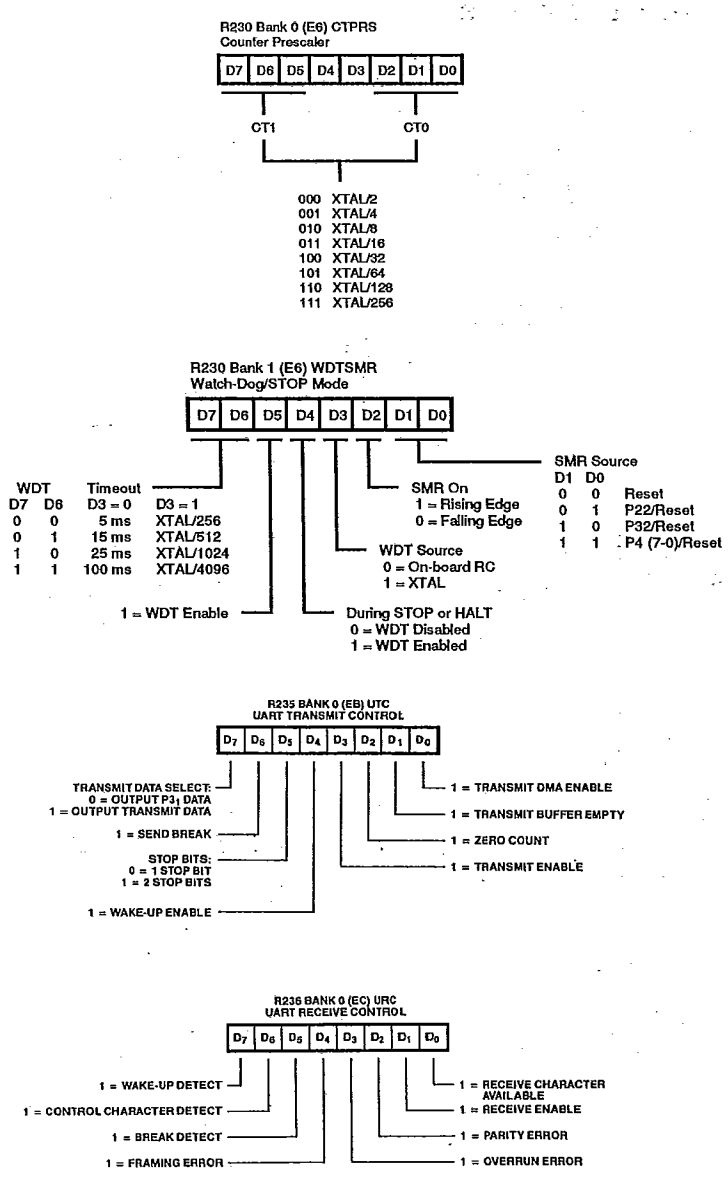


Figure 8. Mode and Control Registers (Continued)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

MODE AND CONTROL REGISTERS (Continued)

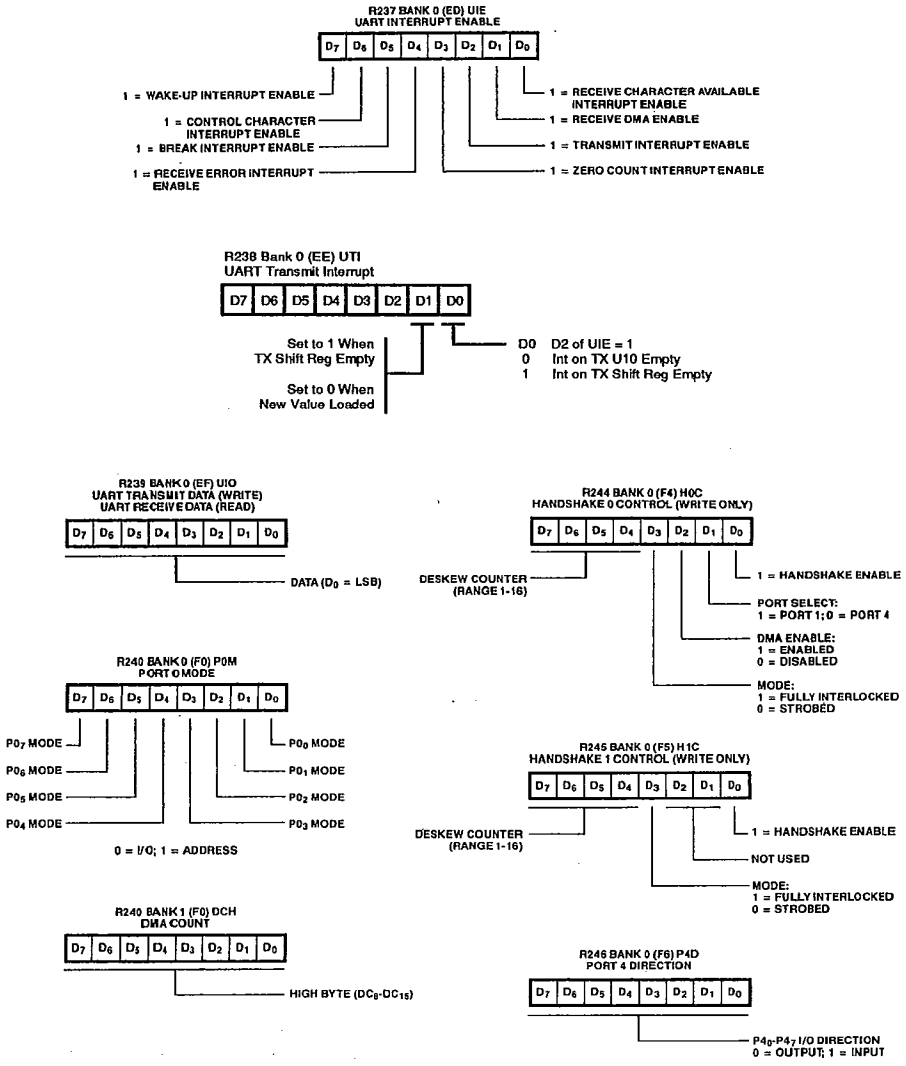


Figure 8. Mode and Control Registers

Powered by ICminer.com Electronic-Library Service CopyRight 2003

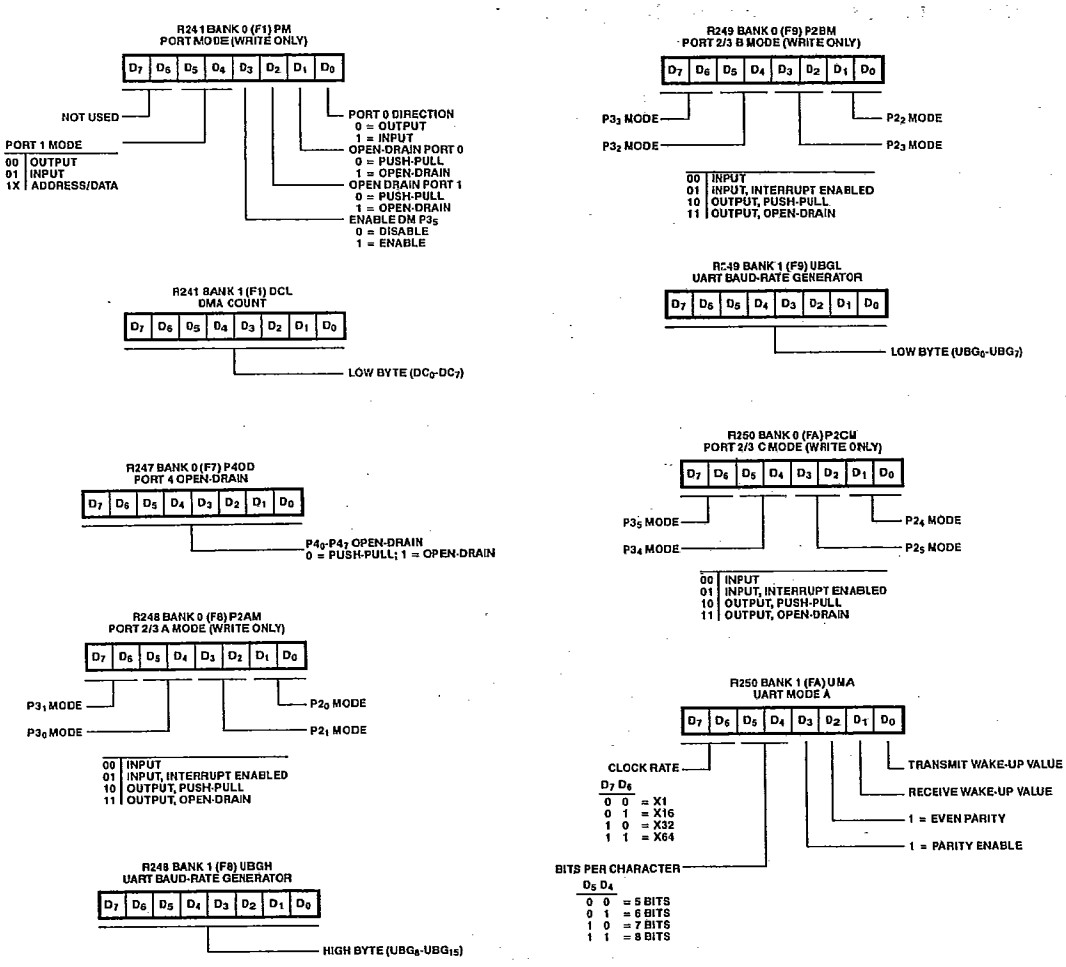


Figure 8. Mode and Control Registers (Continued)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

MODE AND CONTROL REGISTERS (Continued)

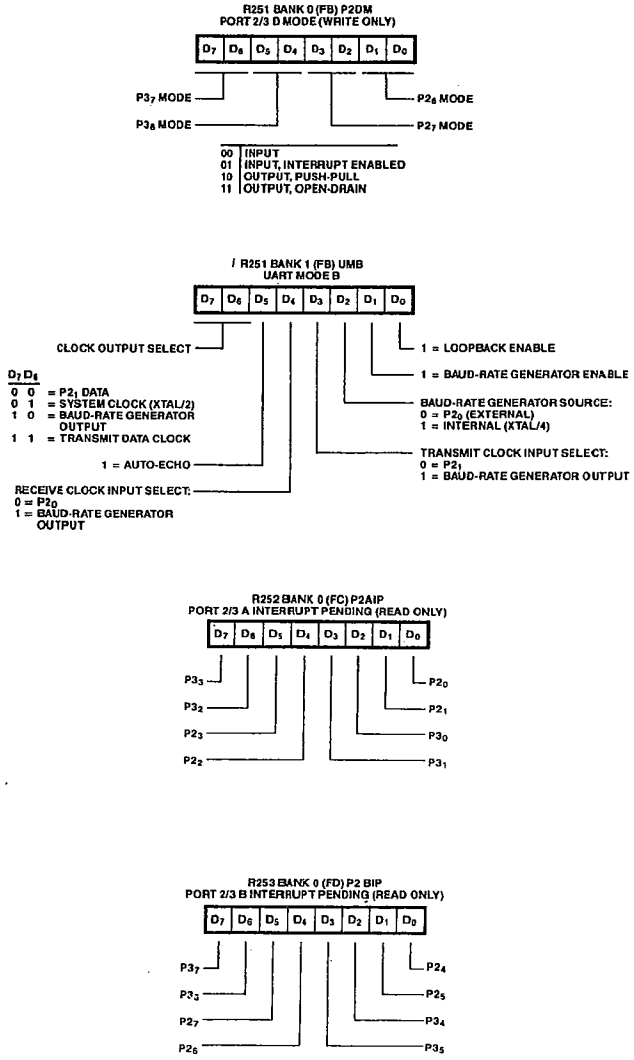


Figure 8. Mode and Control Registers

Powered by ICminer.com Electronic-Library Service CopyRight 2003

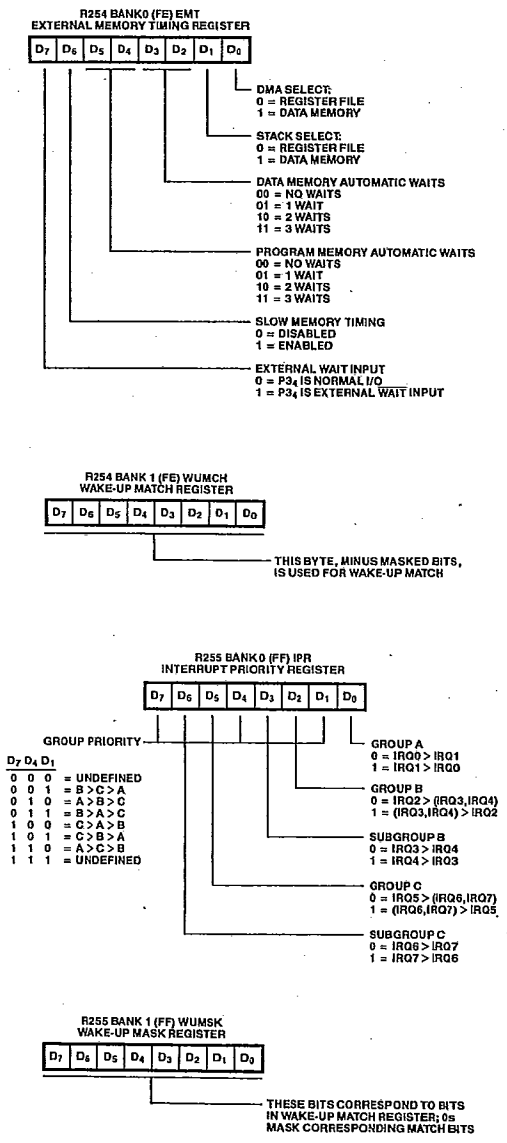


Figure 8. Mode and Control Registers (Continued)

Powered by ICminer.com Electronic-Library Service CopyRight 2003

I/O PORTS

The Super8 has 40 I/O lines arranged into five 8-bit ports. These lines are all TTL-compatible, and can be configured as address/data lines.

Each port has an input register, an output register, and a register address. Data coming into the port is stored in the input register, and data to be written to a port is stored in the output register. Reading a port's register address returns the value in the input register; writing a port's register address loads the value in the output register. If the port is configured for an output, this value will appear on the external pins.

When the CPU reads the bits configured as outputs, the data on the external pins is returned. Under normal output loading, this has the same effect as reading the output register, unless the bits are configured as open-drain outputs.

The ports can be configured as shown in Table 5.

Table 5. Port Configuration

Port	Configuration Choices
0	High address and/or I/O
1	Multiplexed Low address/data or data only.
2 & 3	Control I/O for UART, handshake channels, and counter/timers; also general I/O and external interrupts.
4	Low address or general I/O

Port 0
Port 0 can be configured as an I/O port or an output port for existing external memory, or it can be divided and used as both. The bits configured as I/O can be either all outputs or all inputs; they cannot be mixed. If configured for outputs, they can be push-pull or open-drain type.

Any bits configured for I/O can be accessed via R208. To write to the port, specify R208 as the destination (dst) of an instruction; to read the port, specify R208 as the source (src).

Port 0 bits configured as I/O can be placed under handshake control of handshake channel 1.

Port 0 bits configured as address outputs cannot be accessed via the register, and initially the four lower bits are configured as addresses eight through twelve.

Port 1
Port 1 is configured as a byte-wide address/data port, and in de-mux mode it is a dedicated data bus. Additional address lines can be added by configuring Port 0.

Ports 2 and 3
Ports 2 and 3 provide external control inputs and outputs for the UART, handshake channels, and counter/timers. The pin assignments appear in Table 6.

Bits not used for control I/O can be configured as general purpose I/O lines and/or external interrupt inputs.

Those bits configured for general I/O can be configured individually for input for output. Those configured for output can be individually for input or output. Those configured for output can be individually configured for open drain or push pull output.

All Port 2 and 3 input pins are Schmitt-triggered.

The port address for Port 2 is R210, and for Port 3 is R211.

Table 6. Pin Assignments for Ports 2 and 3

Port 2 Bit	Port 2 Function	Port 3 Bit	Port 3 Function
0	UART receive clock	0	UART receive data
1	UART transmit clock	1	UART transmit data
2	Reserved	2	Reserved
3	Reserved	3	Reserved
4	Handshake 0 input	4	Handshake 1 input/WAIT
5	Handshake 0 output	5	Handshake 1 output/DM
6	Counter 0 input	6	Counter 1 input
7	Counter 0 I/O	7	Counter 1 I/O

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Port 4

Port 4 can be configured as I/O only, or in de-mux mode as the lower address bus only. Each bit can be configured individually as input or output, with either push-pull or open drain outputs. All Port 4 inputs are Schmitt-triggered.

Port 4 can be placed under handshake control handshake channel 0. It's register address is R212.

Precautions on de-multiplexed Z-BUS® Port 4:

1. In de-multiplexed Z-BUS mode Port 4 will be configured as output hence the Port 4 direction register will be ignored.
2. Stop mode recovery from Port 4 will be disabled when opted for a de-multiplexed Z-BUS.
3. The handshake feature using Port 1 or Port 4 will be disabled when opted for de-multiplexed Z-BUS, since Port 1 and Port 4 are configured by default as memory interface.

UART

The UART is a full-duplex asynchronous channel. It transmits and receives independently with 5 to 8 bits per character, and has options for even or odd bit parity, and a wake-up feature.

Data can be read into or out of the UART via R239, Bank 0. This single address is able to serve a full duplex channel because it contains two complete 8-bit registers one for the transmitter and the other for the receiver.

Pins

The UART uses the following Port 2 and 3 pins:

Port/Pin	UART Function
2/0	Receive Clock
3/0	Receive Data
2/1	Transmit Clock
3/1	Transmit Data

Transmitter

When the UART's register address is specified as the destination (dst) of an operation, the data is output on the UART, which automatically adds the start bit, the programmed parity bit, and the programmed number of stop bits. It can also add a wake-up bit if that option is selected.

If the UART is programmed for a 5-, 6-, or 7-bit character, the extra bits in R239 are ignored.

Serial data is transmitted at a rate equal to 1, 1/16, 1/32 or 1/64 of the transmitter clock rate, depending on the programmed data rate. All data is sent out on the failing edge of the clock input.

When the UART has no data to send, it holds the output marking (High). It may be programmed with the Send Break command to hold the output Low (Spacing), which it continues until the command is cleared.

UART Transmit Interrupt Register

The timing for the transmit buffer empty interrupt is software programmable. There are two different interrupt timings selectable with 1 bit.

Option 1: Interrupt is activated at the moment the contents of the TUIO register are transferred to the Tx FIFO.

Option 2: Interrupt is activated at the moment the last stop bit in the Tx FIFO is sent.

After loading the transmit shift register, UART control generates a buffer empty flag to indicate that TUIO is ready to be filled with new data.

A new flag will indicate when the transmit shift register is empty.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

UART (Continued)

D0 of UTI Register
If this bit is zero then a high value of D2 in the UIE register will cause an interrupt on Transmit UIO empty. If this bit is set a high value of D2 in the UIE register will cause an interrupt on transmit shift register empty, that is when the (last) stop bit is transmitted. This bit should be programmed prior to writing to the UIO register. A hardware reset forces this bit to 0.

D1 of UTI Register
This flag is set when the transmit shift register is empty and is reset when a new value is loaded into the UIO. This flag will not be set during a send break. A hardware reset forces this bit to 1. This bit is READ only.

Receiver
The UART begins receive operation when Receive Enable (URC, bit 0) is set High. After this, a Low on the receive input pin for longer than half a bit time is interpreted as a start bit. The UART samples the data on the input pin in the middle of each clock cycle until a complete byte is assembled. This is placed in the Receive Data register.

The 1X clock mode is selected, external bit synchronization must be provided, and the input data is sampled on the rising edge of the clock.

For character lengths of less than eight bits, the UART inserts ones into the unused bits, and if parity is enabled the parity bit is not stripped. The data bits, extra ones and the parity bits are placed in the UART Data register (UIO).

While the UART is assembling a byte in its input shift register, the CPU has time to service an interrupt and manipulate the data character in UIO.

Once complete character is assembled, the UART checks it and performs the following:

If it is an ASCII control character, the UART sets the Control Character status bit.

It checks the wake-up settings and completes any indicated action.

If parity is enabled, the UART checks to see if the calculated parity matches the programmed parity bit. If they do not match, it sets the parity Error bit in URC (R236, Bank 0), which remains set until reset by software.

It resets the Framing Error bit (URC, bit 4) if the character is assembled without any stop bits. This bit remains set until cleared by software.

Overrun errors occur when characters are received faster than they are read. That is, when the UART has assembled a complete character before the CPU has read current character, the UART sets the Overrun Error bit (URC, bit 3), and the character currently in the receive buffer is lost.

The overrun bit remains set until cleared by software.

Address Space
The Super8 can access 64 Kbytes of program memory and 64 Kbytes of data memory. These spaces can be either combined or separate. If separate, they are controlled by the DM line (Port P35), which selects data memory when Low and program memory when High.

Figure 9 shows the system memory space.

CPU Program Memory
Program memory occupies address 0 to 64K. External program memory is accessed by configuring Ports 0 and/or 1 and/or 4 as the memory interface.

The address/data lines are controlled by AS, DS and R/W.

The first 32 program memory bytes are reserved for interrupt vectors; the lowest address available for user programs is 32 (decimal). This value is automatically loaded into the program counter after a hardware reset. Port 0 can be configured to provide from 0 to 8 additional address lines. Port 1 is used as an 8-bit multiplexed address/data port, or as a data port when in de-mux mode.

CPU Data Memory
The external CPU data memory space, if separated from program memory by the DM optional output, can be mapped anywhere from 0 to 64K (full 16-bit address space). Data memory uses the same address/data but (Port 1) and additional address (chosen from Port 0) as program memory. Data memory is distinguished from program memory by the DM pin (P35), and by the fact that data memory can begin at address 0000H. This feature differs from the Z8.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

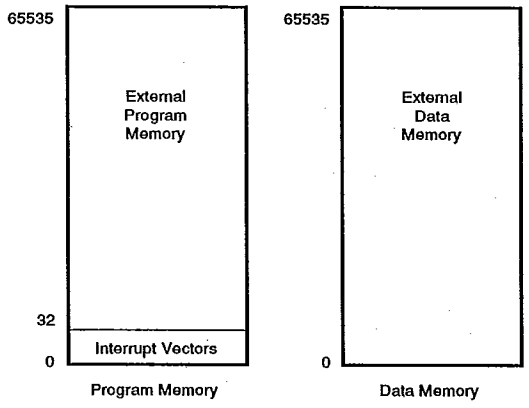


Figure 9. Program and Data Memory Address Space

INSTRUCTION SET

The Super8 instruction set is designed to handle its large register set. The instruction set provides a full complement of 8-bit arithmetic and logical operations, including multiply and divide. It supports BCD operations using a decimal adjustment of binary values, and it supports incrementing and decrementing 16-bit quantities for addressing and counters.

It provides extensive bit manipulation, and rotate and shift operations, and it requires no special I/O instructions - the I/O ports are mapped into the register file.

Instruction Pointer

A special register called the Instruction Pointer (IP) provides hardware support for threaded-code languages. It consists of register-pair R218 and R219, and it contains memory addresses. The MSB is R218.

Threaded-code languages deal with an imaginary higher-level machine within the existing hardware machine. The IP acts like the PC for that machine. The command NEXT passes control to or from the hardware machine to the imaginary machine, and the commands ENTER and EXIT are imaginary machine equivalents of (real machine) CALLS and RETURNS.

If the commands NEXT, ENTER and EXIT are not used, the IP can be used by the fast interrupt processing, as described in the interrupts section.

Flag Register

The flag register (FLAGS) contains eight bits that describe the current status of the Super8. Four of these can be tested and used with conditional jump instructions; two others are used with conditional jump instructions; two others are used for BCD arithmetic, FLAGS also contains the Bank Address bit and the Fast Interrupt Status bit.

The flag bits can be set and reset by instructions.

Caution

Do not specify FLAGS as the destination of an instruction that normally affects the flag bits or the result will be unspecified.

The following paragraphs describe each flag bit:

Bank Address

This bit is used to select one of the register banks (0 or 1) between (decimal) addresses 224 and 255. It is cleared by the SB0 instruction and set by the SB1 instruction.

Fast Interrupt Status

This bit is set during a fast interrupt cycle and reset during the IRET following interrupt servicing. When set, this bit inhabits all interrupts and causes the fast interrupt return to be executed when the IRET instruction is fetched.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTION SET (Continued)

Half-Carry
This bit is set to 1 whenever and addition generates a carry out of bit 3, or when a subtraction borrows out of bit 4. This bit is used by the Decimal Adjust (DA) instruction to convert the binary result of a previous addition or subtraction into the correct decimal (BCD) result. This flag, and the Decimal Adjust flag are not usually accessed by users.

Decimal Adjust
This bit is used to specify what type of instruction was executed last during BCD operations, so a subsequent decimal adjust operation can function correctly. This bit is not usually accessible to programmers, and cannot be used as a test condition.

Overflow Flag
This flag is set to 1 when the result of a two's-complement operation was greater than 127 or less than -128. It is also cleared to 0 during logical operations.

Sign Flag
Following arithmetic, logical, rotate or shift operations, this bit identifies the state of the MSB of the result. A 0 indicates a positive number and a 1 indicates a negative number.

Zero Flag
For arithmetic and logical operations, this flag is set to 1 if the result of the operation is 0.

For operations that test bits in a register, the 0 bit is set to 1 if the result is 0.

For rotate and shift operations, this bit is set to 1 if the result is 0.

Carry Flag
This flag is set to 1 if the result from an arithmetic operation generates carry out of, or a borrow into, bit 7.

After rotate and shift operations, it contains the last value shifted out of the specified register. It can be set, cleared, or complemented by instructions.

Condition Codes
The flags C, Z, S, and V are used to control the operation of conditional jump instructions.

The opcode of a conditional jump contains a 4-bit field called the condition code (cc). The specific under which conditions it is to execute the jump. For example, a conditional jump with the condition code for "equal" after a compare operation only jumps if the two operands are equal.

The condition code and the meanings are given in Table 7.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 7. Condition Codes and Meanings.

Binary	Mnemonic	Flags	Meaning
0000	F	-	Always false
1000	-	-	Always true
0111*	C	C=1	Carry
1111*	NC	C=0	No carry
0110*	Z	Z=1	Zero
1110*	NZ	Z=0	Not zero
1101	PL	S=0	Plus
0101	MI	S=1	Minus
0100	OV	V=1	Overflow
1100	NOV	V=0	No overflow
0110*	EQ	Z=1	Equal
1110*	NE	Z=0	Not equal
1001	GE	(S XOR V)=0	Greater than or equal
0001	LT	(S XOR V)=1	Less than
1010	GT	(Z OR (S XOR V))=0	Greater than
0010	LE	(Z OR (S XOR V))=1	Less than or equal
1111*	UGE	C=0	Unsigned greater than or equal
0111*	ULT	C=1	Unsigned less than
1011	UGT	(C=0 AND Z=0)=1	Unsigned greater than
0011	ULE	(C OR Z)=1	Unsigned less than or equal

Note:
Asterisks (*) indicate condition codes that relate to two different mnemonics but test the same flags. For example, Z and EQ are True if the Zero flag is set, but after an ADD instruction, Z would probably be used, while after a CP instruction, EQ would probably be used.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTION SET (Continued)

Addressing Modes
All operands except for immediate data and condition codes are expressed as register addresses, program memory addresses, or data memory addresses. The addressing modes and the designations are:

- Register (R)
- Indirect Register (IR)
- Indexed (X)
- Direct (DA)
- Relative (RA)
- Immediate (IM)
- Indirect (IA)

Registers can be addressed by an 8-bit address in the range of 0 to 255. Working registers can also be addressed using 4-bit addresses, where five bits contained in a register pointer (R218 or R219) are concatenated with three bits from the 4-bit address to form an 8-bit address.

Registers can be used in pairs to generate 16-bit program or data memory addresses.

Notation and Encoding
The instruction set notations are described in Table 8.

Table 8. Instruction Set Notations

Symbol	Meaning	Symbol	Meaning
r	Working register (between 0 and 15)	DA	Direct address (between 0 and 65535)
rb	Bit of working register	RA	Relative address
r0	Bit 0 of working register	IM	Immediate
R	Register or working register	IML	Immediate long
RR	Register pair or working register pair (Register pairs always start on an even-number boundry)	dst	Destination operand
		src	Source operand
IA	Indirect address	@	Indirect address prefix
Ir	Indirect working register	SP	Stack pointer
IR	Indirect register or indirect working register	PC	Program counter
Irr	Indirect working register pair	IP	Instruction pointer
IRR	Indirect register pair or indirect working register pair	FLAGS	Flags register
		RP	Register pointer
X	Indexed	#	Immediate operand prefix
XS	Indexed, short offset	%	Hexadecimal number prefix
XL	Indexed, long offset	OPC	Opcode

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Functional Summary of Commands

Figure 10 shows the formats followed by a quick reference guide to the commands.

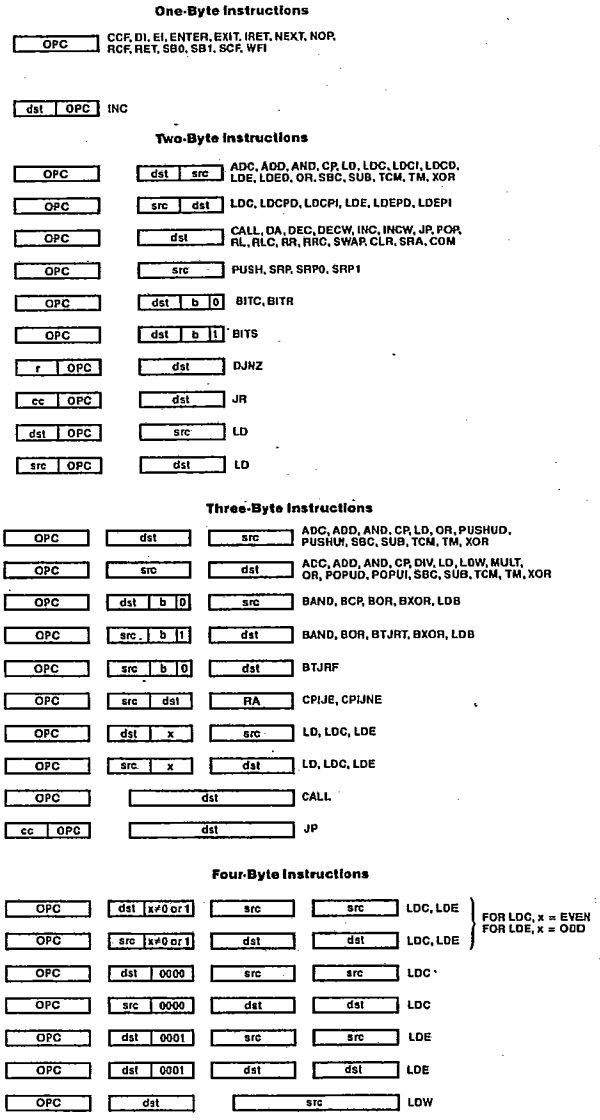


Figure 10. Instruction Formats

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTION SUMMARY

Note: 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 operand location. For example:

dst (7)

refers to bit 7 of the destination operand.

Instruction and Operation	Address Mode dst	src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
ADC dst, src dst←dst + src +C	†		1[ ]	*	*	*	-	0	*
ADD dst, src dst←dst + src	†		0[ ]	*	*	*	*	0	*
AND dst, src dst←dst AND src	†		5[ ]	-	*	*	0	-	-
BAND dst, src dst←dst AND src	r0 Rb	Rb r0	67 67	-	*	0	U	-	-
BCP dst, src dst – src	r0	Rb	17	-	*	0	U	-	-
BITC dst dst←NOT dst	rb		57	-	*	0	U	-	-
BITR dst dst←0	rb	77	-	-	-	-	-	-	-
BITS dst dst←0 dst←1	rb	77	-	-	-	-	-	-	-
BOR dst, src dst←dst OR src	r0	rB	07	-	*	0	U	-	-
BTJRF dst←0 if src=0, PC=PC+dst	RA	rb	37	-	-	-	-	-	-
BTJRT if src=1, PC=PC+dst	RA	rb	37	-	-	-	-	-	
BXOR dst, src dst←dst XOR src	r0	Rb	27	-	*	0	U	-	-
CALL dst SP←SP - 2 @SP←PC, PC←dst	DA IRR IA		F6 F4 D4	-	-	-	-	-	-
CCF C←NOT C			EF	*	-	-	-	-	-

Instruction and Operation	Address Mode dst	src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
CLR dst dst←0	R IR		B0 B1	-	-	-	-	-	-
COM dst dst←NOT dst	R IR		60 61	-	*	*	0	-	-
CP dst, src dst - src	†		A[ ]	*	*	*	*	-	-
CPIJE if dst - src=0, then PC ←PC+RA Ir ←Ir +1	r	Ir	C2	-	-	-	-	-	-
CPIJNE if dst - src=0, then PC ←PC+RA Ir ←Ir +1	r	Ir	D2	-	-	-	-	-	-
DA dst dst←DA dst	R IR		40 41	*	*	*	U	-	-
DEC dst dst←dst - 1	R IR		00 01	-	*	*	*	-	-
DECW dst dst←dst - 1	RR IR		80 81	-	*	*	*	-	-
DI SMR(0)←0			8F	-	-	-	-	-	-
DIV dst, src dst÷src dst (Upper)← Quotient dst (Lower)← Remainder	RR RR RR	R IR IM	94 95 96	*	*	*	*	-	-
DJNZ r, dst r←r - 1 if r = 0 PC←PC+dst	RA	r	rA (r=0 toF)	-	-	-	-	-	-
EI SMR(0)			9F	-	-	-	-	-	-

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTION SUMMARY (Continued)

Instruction and Operation	Address Mode dst	src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
ENTER SP←SP - 2 @ SP←IP IP←PC PC←@ IP IP←IP + 2			1F	-	-	-	-	-	-
EXIT IP←@SP SP←SP + 2 PC←@ IP IP←IP + 2			2F	-	-	-	-	-	-
INC dst dst←dst + 1	r		rE r = 0 - F	-	*	*	*	-	-
	R		20						
	IR		21						
INCW dst dst←dst + 1	RR		A0	-	*	*	*	-	-
	IR		A1						
IRET (Fast) PC↔IP FLAG←FLAG FIS←0			BF	Restored to before interrupt					
IRET (Normal) FLAGS←@SP; SP←SP + 1; PC←@SP; SP←SP + 2; SMR(0)←1			BF	Restored to before interrupt					
JP cc, dst if cc is true, PC←dst	DA		ccD c = 0 to F	-	-	-	-	-	-
	IRR		30						
JR cc, dst if cc is true, PC←PC + d	RA		ccB cc = 0 to F	-	-	-	-	-	-
LD dst, src dst←src	r	IM	rC	-	-	-	-	-	-
	r	R	r8						
	R	r	r9						
			r = 0 to F						
	r	IR	C7						
	IR	r	D7						
	R	R	E4						
	R	IR	E5						
	R	IM	E6						
	IR	IM	D6						
	IR	R	F6						
	r	x	87						
	x	r	97						

Instruction and Operation	Address Mode dst	src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
LDB dst, src dst←src	r0	Rb	47	-	-	-	-	-	-
	Rb	r0	47						
LDC/LDE dst←src	r	Irr	C3	-	-	-	-	-	-
	Irr	r	D3						
	r	xs	E7						
	xs	r	F7						
	r	x1	A7						
	x1	r	B7						
	r	DA	A7						
	DA	r	B7						
LDCD/LDED dst, src dst←src rr←rr − 1	r	Irr	E2	-	-	-	-	-	-
LDEI/LDCI dst, src dst←src rr←rr + 1	r	Irr	E3	-	-	-	-	-	-
LDCPD/LDCI dst, src dst←src rr←r + 1	r	Irr	E3	-	-	-	-	-	-
LDCPI/LDEPI dst, src rr←rr + 1 dst←src	Irr	r	F3	-	-	-	-	-	-
LDW dst, src dst←src	RR	RR	C4	-	-	-	-	-	-
	RR	IR	C5						
	RR	IMM	C6						
MULT dst, src	RR	R	84	*	0	*	*	-	-
	RR	IR	85						
	RR	IM	86						
NEXT PC←@ IP IP←IP + 2			0F	-	-	-	-	-	-
NOP			FF	-	-	-	-	-	-
OR dst, src dst←dst OR src	†		4[ ]	-	*	*	0	-	-
POP dst dst←@SP; SP←SP + 1		R	50	-	-	-	-	-	-
		IR	51						
POPUD dst, src dst←src IR←IR - 1	R	IR	92	-	-	-	-	-	-
POPUI dst, src dst←src IR←IR + 1	R	IR	93	-	-	-	-	-	-

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTION SUMMARY (Continued)

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
PUSH src SP←SP - 1; @SP←src		R IR	70 71	-	-	-	-	-	-
PUSHUD dst, src IR←IR - 1 dst←src	IR	R	82	-	-	-	-	-	-
PUSHUI dst, src IR←IR + 1 dst←src	IR	R	83	-	-	-	-	-	-
RCF C←0			CF	0	-	-	-	-	-
RET PC←@SP;SP←SP+2			AF	-	-	-	-	-	-
RL dst C←dst(7) dst(0)←dst(7) dst(N+1)←dst(N) N=0 to 6	R IR		90 91	*	*	*	*	-	-
RLC dst dst(0)←C C←dst(7) dst(N+1)←dst(N) N=0 to 6	R IR		10 11	*	*	*	*	-	-
RR dst C←dst(0) dst(7)←dst(0) dst(N)←dst(N+1) N=0 to 6	R IR		E0 E1	*	*	*	*	-	-
RRC dst C←dst(0) dst(7)←C dst(N)←dst(N+1) N=0 to 6	R IR		C0 C1	*	*	*	*	-	-

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
SB0 BANK←0			4F	-	-	-	-	-	-
SB1 BANK←1			5F	-	-	-	-	-	-
SBC dst,src dst←dst - scr - C	†		3[ ]	*	*	*	*	1	*
SCF C←1			DF	1	-	-	-	-	-
SRA dst dst(7)←dst(7) C←dst(0) dst(N)←dst(N+1) N=0 to 6	R IR	D1	D0	*	*	*	0	-	-
SRP src RP0←IM RP1←IM+8		IM	31	-	-	-	-	-	-
SRP0 RP0←IM		IM	31	-	-	-	-	-	-
SRP1 RP1←IM		IM	31	-	-	-	-	-	-
STOP			6F	-	-	-	-	-	-
SUB dst,src dst←dst - src	†		2[ ]	*	*	*	*	1	*
SWAP dst dst(0-3)↔dst(4-7)	R IR		F0 F1	-	*	*	U	-	-
TCM dst, src (NOT dst) AND src	†		6[ ]	-	*	*	0	-	-
TM dst, src dst AND src	†		7[ ]	-	*	*	0	-	-

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
TSW dst, src	R	R	7F	U	*	*	0	U	U
WFI			3F	-	-	-	-	-	-
XOR dst, src dst←dst XOR src	†		B[ ]	-	*	*	0	-	-

† 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.

Address Mode dst	Address Mode src	Lower Opcode Nibble
r	r	[2]
r	Ir	[3]
R	R	[4]
R	IR	[5]
R	IM	[6]

Notes:
0 = Cleared to Zero
1 = Set to One
- = Unaffected
* = Set or reset, depending on result of operation.
U = Undefined

Powered by ICminer.com Electronic-Library Service CopyRight 2003

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 DEC R_1	6 DEC IR_1	6 ADD r_1,r_2	6 ADD r_1,Ir_2	10 ADD R_2,R_1	10 ADD IR_2,R_1	10 ADD R_1,IM	10 BOR* r_0-R_b	6 LD r_1,R_2	6 LD r_2,R_1	12/10 DJNZ r_1,RA	12/10 JR cc,RA	6 LD r_1,IM	12/10 JP cc,DA	6 INC r1	14 NEXT
1	6 RLC R_1	6 RLC IR_1	6 ADC r_1,r_2	6 ADC r_1,Ir_2	10 ADC R_2,R_1	10 ADC IR_2,R_1	10 ADC R_1,IM	10 BCP r_1,b,R_2								20 ENTER
2	6 INC R_1	6 INC IR_1	6 SUB r_1,r_2	6 SUB r_1,Ir_2	10 SUB R_2,R_1	10 SUB IR_2,R_1	10 SUB R_1,IM	10 BXOR* r_0-R_b								22 EXIT
3	10 JP IRR_1	NOTE C	6 SBC r_1,r_2	6 SBC r_1,Ir_2	10 SBC R_2,R_1	10 SBC IR_2,R_1	10 SBC R_1,IM	NOTE A								6 WFI
4	6 DA R_1	6 DA IR_1	6 OR r_1,r_2	6 OR r_1,Ir_2	10 OR R_2,R_1	10 OR IR_2,R_1	10 OR R_1,IM	10 LDB* r_0-R_b								6 SBO
5	10 POP R_1	10 POP IR_1	6 AND r_1,r_2	6 AND r_1,Ir_2	10 AND R_2,R_1	10 AND IR_2,R_1	10 AND R_1,IM	8 BITC r_1,b								6 SBI
6	6 COM R_1	6 COM IR_1	6 TCM r_1,r_2	6 TCM r_1,Ir_2	10 TCM R_2,R_1	10 TCM IR_2,R_1	10 TCM R_1,IM	10 BAND* r_0-R_b								6 STOP
7	10/12 PUSH R_2	12/14 PUSH IR_2	6 TM r_1,r_2	6 TM r_1,Ir_2	10 TM R_2,R_1	10 TM IR_2,R_1	10 TM R_1,IM	NOTE B								10 TSW RR
8	10 DECW RR_1	10 DECW IR_1	10 PUSHUD IR_1,R_2	10 PUSHUI IR_1,R_2	24 MULT R_2,RR_1	24 MULT IR_2,RR_1	24 MULT IM,RR_1	10 LD r_1,x,r_2								6 DI
9	6 RL R_1	6 RL IR_1	10 POPUD IR_2,R_1	10 POPUI IR_2,R_1	28/12 DIV R_2,RR_1	28/12 DIV IR_2,RR_1	28/12 DIV IM,RR_1	10 LD r_2,x,r_1								6 EI
A	10 INCW RR_1	10 INCW IR_1	6 CP r_1,r_2	6 CP r_1,Ir_2	10 CP R_2,R_1	10 CP IR_2,R_1	10 CP R_1,IM	NOTE D								14 RET
B	6 CLR R_1	6 CLR IR_1	6 XOR r_1,r_2	6 XOR r_1,Ir_2	10 XOR R_2,R_1	10 XOR IR_2,R_1	10 XOR R_1,IM	NOTE E								16/6 IRET
C	6 RRC R_1	6 RRC IR_1	16/18 CPIJE Ir_2,RA	12 LDC* r_1,Irr_2	10 LDW RR_2,RR_1	10 LDW IR_2,RR_1	12 LDW RR_1,IML	6 LD r_1,Ir_2								6 RCF
D	6 SRA R_1	6 SRA IR_1	16/18 CPIJNE Ir_1,r_2,RA	12 LDC* r_2,Irr_1	20 CALL IA_1		10 LD IR_1,IM	6 LD Ir_1,r_2								6 SCF
E	6 RR R_1	6 RR IR_1	16 LDCD* r_1,Irr_2	16 LDCI* r_1,Irr_2	10 LD R_2,R_1	10 LD IR_2,R_1	10 LD R_1,IM	18 LDC* r_1,Irr_2.xs								6 CCF
F	8 SWAP R_1	8 SWAP IR_1	16 LDCPD* r_2,Irr_1	16 LDCPI* r_2,Irr_1	18 CALL IRR_1	10 LD R_2,IR_1	18 CALL DA_1	18 LDC* r_2,Irr_1.xs								6 NOP

NOTE A

16/18 BTJRF r_2,b,RA	16/18 BTJRT r_2,b,RA

NOTE B

8 BITR r_1,b	8 BITS r_1,b

NOTE C

6 SRP IM	6 SRP0 IM	6 SRP1 IM

NOTE D

20 LDC* r_1,Irr_2,xL	20 LDC* r_1,DA_2

NOTE E

20 LDC* r_2,Irr_2,xL	20 LDC* r_2,DA_1

Legend:
r = 4-bit address
R = 8-bit address
b = bit number
R_1 or r_1 = dst address
R_2 or r_2 = src address

***Examples:**
BOR r_0-R_2
is BOR r_1,b,R_2
or BOR r_2,b,R_1
LDC r_1,Irr_2
is LDC r_1,Irr_2 = program
or LDE r_1,Irr_2 = data

Sequence:
Opcode, first, second, third operands

NOTE The blank areas are not defined.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INSTRUCTIONS

Table 9. Super8 Instructions

Mnemonic	Operands	Instructions
Load Instructions		
CLR	dst	Clear
LD	dst, src	Load
LDB	dst, src	Load bit
LDC	dst, src	Load program memory
LDE	dst, src	Load data memory
LDCD	dst, src	Load program memory and decrement
LDED	dst, src	Load data memory and decrement
LDCI	dst, src	Load program memory increment
LDEI	dst, src	Load data memory and increment
LDCPD	dst, src	Load program memory with pre-decrement
LDEPD	dst, src	Load data memory with pre-decrement
LDCPI	dst, src	Load program memory with pre-increment
LDEPI	dst, src	Load data memory with pre-increment
LDW	dst, src	Load word
POP	dst	Pop stack
POPUD	dst, src	Pop user stack (decrement)
POPUI	dst, src	Pop user stack (increment)
PUSH	src	Push stack
PUSHUD	dst, src	Push user stack (decrement)
PUSHUI	dst, src	Push user stack (increment)
Arithmetic Instructions		
ADC	dst, src	Add with carry
ADD	dst, src	Add
CP	dst, src	Compare
DA	dst	Decimal adjust
DEC	dst	Decrement
DECW	dst	Decrement word
DIV	dst, src	Divide
INC	dst	Increment
INCW	dst	Increment word
MULT	dst, src	Multiply
SBC	dst, src	Subtract with carry
SUB	dst, src	Subtract
Logical Instructions		
AND	dst, src	Logical AND
COM	dst	Complement
OR	dst, src	Logical OR
XOR	dst, src	Logical exclusive
Program Control Instructions		
BTJRT	dst, src	Bit test jump relative on True
BTJRF	dst, src	Bit test jump relative on False
CALL	dst	Call procedure
CPIJE	dst, src	Compare, increment and jump on equal
CPIJNE	dst, src	Compare, increment and jump on non-equal
DJNZ	r, dst	Decrement and jump on non-zero
ENTER		Enter

Mnemonic	Operands	Instructions
Program Control Instructions (Continued)		
EXIT		Exit
IRET		Return from interrupt
JP	cc, dst	Jump on condition code
JP	dst	Jump unconditional
JR	cc, dst	Jump relative on condition code
JR	dst	Jump relative unconditional
NEXT		Next
RET		Return
WFI		Wait for interrupt
Bit Manipulation Instructions		
BAND	dst, src	Bit AND
BCP	dst, src	Bit compare
BITC	dst	Bit complement
BITR	dst	Bit reset
BITS	dst	Bit set
BOR	dst, src	Bit OR
BXOR	dst, src	Bit exclusive OR
TCM	dst, src	Test complement under mask
TM	dst, src	Test under mask
TSW	src1,src2	Test Word
Rotate and Shift Instructions		
RL	dst	Rotate left
RLC	dst	Rotate left through carry
RR	dst	Rotate right
RRC	dst	Rotate right through carry
SWAP	dst	Swap nibbles
CPU Control Instructions		
CCF		Complement carry flag
DI		Disable interrupts
EI		Enable interrupts
NOP		Do nothing
RCF		Reset carry flag
SB0		Set bank 0
SB1		Set bank 1
SCF		Set carry flag
SRP	src	Set register pointers
SRP0	src	Set register pointer zero
SRP1	src	Set register pointer one
STOP		Enable STOP Mode

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INTERRUPTS

The Super8 interrupt structure contains 8 levels of interrupt, 16 vectors and 27 sources.

Interrupt priority is assigned by level, controlled by the Interrupt Priority Register (IPR). Each level is masked (or enabled) according to the bits in the Interrupt Mask Register (MR), and the entire interrupt structure can be disabled by clearing a bit in the System Mode Register (R222).

The major components of the interrupt structure are sources, vectors and levels. These are shown in Figure 10 and discussed in the following paragraphs.

Sources

A source is anything that generates an interrupt. This can be internal or external to the Super8 MCU. Internal sources are hardwired to a particular vector and level, while external sources can be assigned to various external events. External interrupts are falling-edge triggered.

Vectors

The 16 vectors are divided unequally among the eight levels. For example, vector 12 belongs to level 2, while level 3 contains vectors 0, 2, 4, and 6.

The vector number is used to generate the address of a particular interrupt servicing routine; therefore all interrupts using the same vector must use the same interrupt handling routine.

Levels

Levels provide the top level of priority assignment. While the sources and vectors are hardwired within each level, the priorities of the levels can be changed by using the Interrupt Priority Register (see Figure 8 for bit details).

If more than one interrupt source is active, the source from the highest priority level will be serviced first. If both sources are from the same level, the source with the lowest vector will have priority. For example, if the UART Receive Data bit and UART Parity Error bit are both active, the UART Parity Error bit will be serviced first because it is vector 16, and UART received data is vector 20.

The levels are shown in Figure 12.

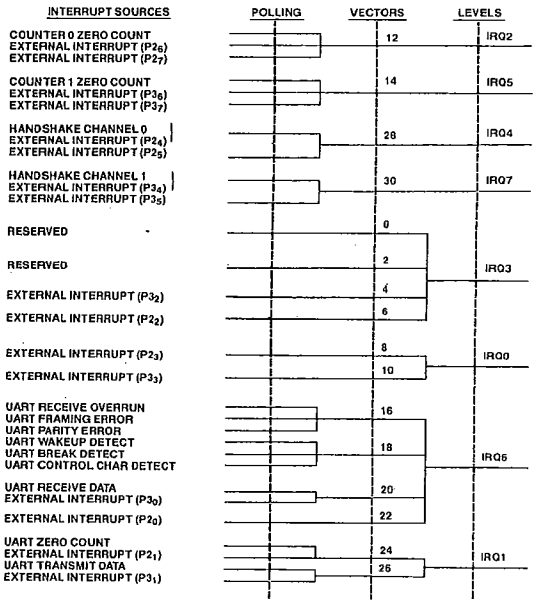


Figure 12. Interrupt Levels and Vectors

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Enables
Interrupts can be enabled or disabled as follows:

- **Interrupt enable/disable.** The entire interrupt structure can be enabled or disabled by setting bit 0 in the System Mode register (R222).
- **Level enable.** Each level can be enabled or disabled by setting the appropriate bit in the Interrupt Mask register (R221).
- **Level priority.** The priority of each level can be controlled by the values in the Interrupt Priority register (R255, Bank 0).
- **Source enable/disable.** Each interrupt source can be enabled or disabled in the source's Mode and Control register.

Source Routines
Before an interrupt request can be granted a) interrupts must be enabled, b) the level must be enabled, c) it must be the highest priority interrupting level, d) it must be enabled at the interrupting source, and) it must have the highest priority within the level.

If this all occurs, an interrupt request is granted.

The Super8 then enters an interrupt machine cycle that completes the following sequence:

- It resets the Interrupt Enable bit to disable all subsequent interrupts.
- It saves the Program Counter and status flags on the stack.
- It branches to the address contained within the vector location for the interrupt.
- It passes control to the interrupt servicing routine.

When the interrupt servicing routine has serviced the interrupt, it should issue an interrupt return (RET) instruction. This restores the Program Counter and status flags and sets the Interrupt Enable bit in the System Mode Register.

Fast Interrupt Processing
The Super8 provides a feature called fast interrupt processing, which completes the interrupt servicing in 6 clock periods instead of the usual 22.

Two hardware registers support fast interrupts. The Instruction Pointer (IP) holds the starting address of the service routine, and saves the PC value when a fast interrupt occurs. A dedicated register, FLAG, saves the contents of the FLAGS register when a fast interrupt occurs.

To use this feature, load the address of the service routine in the Instruction Pointer, load the level number into the Fast Interrupt Select field, and turn on the Fast Interrupt Enable bit in the System Mode Register.

When an interrupt occurs in the level selected for fast interrupt processing, the following occurs:

- The contents of the Instruction Pointer and Program Counter are swapped.
- The contents of the Flag register are copied into FLAG.
- The fast interrupt Status Bit in FLAGS is set.
- The interrupt is serviced.
- When IRET is issued after the interrupt service outline is completed, the Instruction Pointer and Program Counter are swapped again.
- The contents of FLAG are copied back into the Flag register.
- The Fast Interrupt Status bit in FLAGS is cleared.

The interrupt servicing routine selected for fast processing should be written so that the location after the IRET instruction is the entry point the next time the (same) routine is used.

Level or Edge Triggered
Because internal interrupt requests are levels and interrupt requests from the outside are (usually) edges, the hardware for external interrupts uses edge-trimmed flip-flops to convert the edges to levels.

The level-activated system requires that interrupt-servicing software perform some action to remove the interrupting source. The action involved in serving the interrupt may remove the source, or the software may have to actually reset the flip-flops by writing to the corresponding Interrupt Pending Register.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

STACK OPERATION

The Super8 architecture supports stack operations in the register file or in data memory. Bit 1 in the external Memory Timing register (R254, Bank 0) selects between the two.

Register pair 216-217 forms the Stack Pointer used for all stack operations. R216 is the MSB and R217 is the LSB.

The Stack Pointer always points to data stored on the tip of the stack. The address is decremented prior to a PUSH and incremented after a POP.

The stack is also used as a return stack for CALLs and interrupts. During a CALL, the contents of the PC are saved on the stack, to be restored later. Interrupts cause the contents of the PC and FLAGS to be saved on the stack for recovery by IRET when the interrupt is finished.

When the Super8 is configured for an internal stack (using the register file), R217 contains the Stack Pointer. R216 may be used as a general-purpose register, but its contents will be changed if an overflow or underflow occurs as the result of incrementing or decrementing the stack address during normal stack operations.

User-Defined Stacks

The Super8 provides for user-defined stacks in both the register file and program or data memory. These can be made to increment or decrement on a push by the choice of opcodes. For example, to implement a stack that goes from Low addresses to High addresses in the register file, use PUSHUI and POPUD. For a stack that goes from High address to Low addresses in data memory, use LDEI for POP and LDEPD for PUSH.

COUNTER/TIMERS

The Super8 has two identical, independently programmable, 16-bit counter/timers that can be cascaded to produce a single 32-bit counter. They can be used to count external events, or they can obtain their input internally. The internal input is obtained by dividing the crystal frequency by four.

The counter/timers can be set to count up or down, by software or external events. They can be set for single or continuous cycle counting, and they can be set with a bi-value option, where two preset time constants alternate in loading the counter each time it reaches zero. This can only be used to produce an output pulse train with a variable duty cycle.

The counter/timers can also be programmed to capture the count value at an external event or generate an interrupt whenever the count reaches zero. They can be turned on and off in response to external events by using a gate and/or a trigger option. The gate option enables counts only when the gate line is Low; the trigger option turns on the counter after a transient High. The gate and trigger options used together causes the counter/timer to work in gate mode after initially being triggered.

The control and status register bits for the counter/timers are shown in Figure 5.

DMA

The Super8 features an on-chip Direct Memory Access (DMA) channel to provide high bandwidth data transmission capabilities. The DMA channel can be used by the UART receiver, UART transmitter, or handshake channel 0. Data can be transferred between the peripheral and contiguous locations in either the register file or external data memory. A 16-bit count register determines the number of transactions to be performed; an interrupt can be generated when the count is exhausted. DMA transfers to or from the register file require six CPU clock cycles; DMA transfers to or from external memory takes ten CPU clock cycles, excluding wait states.

WDT

The WDT is initially enabled by writing a "1" to bit D5 in the WDT/SMR (see Figure 8) register and re-triggered on subsequent writings to the same bit. The timer circuit is driven by two selectable sources. An internal RC oscillator can be selected resulting in four different time-out periods (5 ms - 15 ms - 25 ms - 100 ms) or the XTAL oscillator can be selected. The actual RC time-out periods should be no more than three times greater than the aforementioned, but for specification purposes, the aforementioned times will be used. The WDT will default to an "off" or disable condition. The RESET pin will be held Low for 5 ms after a WDT time out.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

STOP MODE

A stop instruction is added to enable the Super8 to go in a stop mode. In a STOP mode the internal system clock and the external crystal oscillator are disabled. Stop Mode Recovery is achieved by two software programmable sources:

Reset. *Stop Mode Recovery using reset.* A hardware reset will restart the XTAL oscillator and reset the entire chip. The reset signal to recover from STOP should be longer than a normal reset pulse since the oscillator has to start up.

Input. *Stop Mode Recovery using input.* Edge-triggering of a predefined input on Port 2, Port 3 or Port 4 will restart the oscillator but won't reset the part. The Super8 will proceed where it was stopped, after the oscillator has been stabilized.

If the input used to recover is interrupt enabled, the Super8 will generate an interrupt after the oscillator is stable.

SMR using reset will configure all I/O ports (like a normal hardware reset) as inputs. This causes I/O lines, which were driving some peripherals prior to invoking STOP mode, to lose their drive since the I/O line becomes an input. The weak latch on the I/O line is not powerful enough to provide high drive.

Every STOP instruction should be proceeded by a NOP instruction, since this instruction is activated one machine cycle earlier than any other instruction.

WDT/SMR REGISTER

New Register: E6 Bank 1 WDT/SMR

D1, D0 Stop Mode Recovery source select.
Bit D0 and D1 determine the Stop Mode Recovery source.

D1	DO	
0	0	Recovery from RESET only
0	1	Recovery from P22 and RESET
1	0	Recovery from P32 and RESET
1	1	Recovery from any input of Port 4 and RESET

A hardware reset forces D0 and D1 to zero.

D2 Stop recovery edge.
A 1 in this position indicates that a rising edge on any one of the recovery sources wakes the Super8 from Stop mode. A 0 indicates falling edge recovery. The reset value is 0.

D3 XTAL1/RC Select for WDT.
When a zero is written to D3, the clock of the WDT is driven by the on-board RC oscillator. If D3 is set to 1 the WDT is driven by XTAL1. D3 has a zero reset value.

D4 WDT enable during STOP or HALT.
When this bit is set WDT is enabled during STOP or HALT. In this case recovery from STOP or HALT should be done before the selected time-out.

A zero in this bit location will disable the WDT while the Super8 is stopped or halted. A hardware reset forces this bit to a zero.

D5 WDT
The Watch-Dog Timer is initially enabled by writing a 1 to D5 and retriggered on subsequent writings to the same bit. Reset value = 0. Writing a zero to this bit has no effect. Once a 1 is written to D5, it persists until a hardware reset occurs.

D6, D7 WDT Time-Out
Depending on the logical state of these bits, two sets of four different time-out values can be selected.

A normal reset signal must be active Low during 5 XTAL clock periods. Using the reset signal input to recover from STOP mode requires 10 XTAL clock periods in order to let XTAL oscillation start up and stabilize, generating a good oscillator output level. During WDT timer time out the reset pin is held Low in source to accomplish a system reset with other peripherals of the Super8. When the reset pin is held Low, the capability of sink current via the reset pin should be taken into account (see DC Characteristics).

Powered by ICminer.com Electronic-Library Service CopyRight 2003

HALT MODE REGISTER

HALT mode is invoked by executing the WFI instruction when bit D3 of the HALT mode register is loaded with a zero. When the Super8 is in HALT mode the internal CPU clock is disabled. The user has the choice to leave the UART or the timers active or not active during halt. Recovery from HALT occurs on an interrupt request or DMA request. The HALT mode makes it possible to support low operating current.

D0 CT0 in HALT mode.
Writing a zero to this bit will disable the CT0 in HALT mode. No interrupt request will be generated in this case. A 1 will keep the CT0 active. A hardware reset forces this bit to a 1.

D1 CT1 in HALT mode.
Similar to CT0. When the counters are cascaded, the HALT mode 32-bit counter is determined by the logical state of D1. A hardware reset forces this bit to a 1.

D2 UART HALT mode.
Writing a zero to the bit will disable the UART. No interrupt request will be generated. A 1 will make the UART and its interrupt logic remain active in HALT mode. A hardware reset forces this bit to a 1.

D3 CPU HALT mode.
Writing a zero to this bit will invoke the HALT mode upon the execution of the WFI instruction. The UART and counters only can be halted if D3 is 0. During HALT the internal CPU clock is disabled, and no address strobe is generated. A hardware reset sets this bit to a 1.

Counter/Timer prescaler options.
Current Z88C00 has XTAL/4 prescaler only. An 8-bit prescaler for each counter will be implemented so there will be three bits needed in a new register to select the six prescalers.

XTAL/2 (N/A for counter mode)
XTAL/4 (default value)
XTAL/8
XTAL/16
XTAL/32
XTAL/64
XTAL/128
XTAL/256

When the counters are cascaded, only the prescaler of CT1 is activated.

A hardware reset loads this register with 21 (hex).

Above new registers are all Read/Write.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

ABSOLUTE MAXIMUM RATINGS

Symbol	Description	Min	Max	Unit
V_{DD}	Supply Voltage*	-0.3	+7.0	V
T_{STG}	Storage Temp	-65	+150	C
T_A	Oper Ambient Temp	†	†	C

* Voltages on all pins with respect to GND.
† See Ordering Information

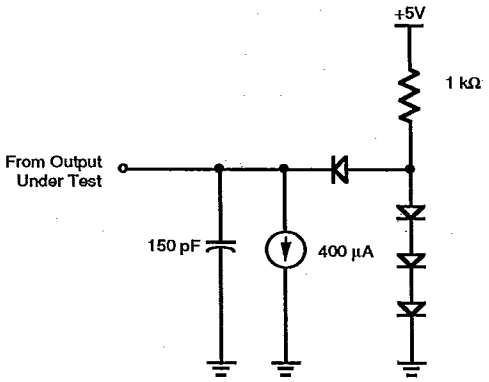
Stress greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating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

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to V_{SS}. Positive current flows into the referenced pin (Standard Test Load).

Standard conditions are:

- $4.5V < V_{CC} < 5.5V$
- GND - 0V
- $-40°C < T_A < +105°C$



Standard Test Load

Powered by ICminer.com Electronic-Library Service CopyRight 2003

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	2.2	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}=-400 µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL}=+4.0 mA
V_{IL}	Input Leakage	-10	10	µA	
I_{OL}	Output Leakage	-10	10	µA	
I_{IR}	Reset Input Current		-50	µA	
I_{CC}	V_{CC} Supply Current		90	mA	[1]
I_{CC1}	Standby Current		5	mA	@ 20 MHz [2]
			10	mA	@ 30 MHz [2]
I_{CC2}	Standby Current		20	µA	[3]

Notes:
Estimated Values
[1] In this case all outputs and I/O pins are floating.
[2] HALT mode is invoked with UART CT0 and CT1 deactivated with all input pins tied to V_{CC} or V_{SS}.
[3] STOP mode is invoked with all input pins tied to V_{CC} or V_{SS}.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

AC ELECTRICAL CHARACTERISTICS

External I/O or Memory Read and Write Timing

Number	Symbol	Parameter
1	TdA(AS)	Address valid to /AS Rise Delay
2	ThAS(A)	/AS Rise to Address Valid Hold Time
3	TdAS(DI)	/AS Rise to Data In Required Valid Delay
4	TwAS	/AS Low Width
5	TdAZ (DSR)	Address Float to /DS (Read)
6	TwDSR	/DS (Read) Low Width
7	TwDSW	/DS (Write) Low Width
8	TdDSR (DI)	/DS (Read) to Data In Required Valid Delay
9	ThDSR(DI)	/DS Rise (Read) to Data In Hold Time
10	TdDS (A)	/DS Rise to Address Active Delay
11	TdDA (AS)	/DS Rise to /AS Delay
12	TdR/W (AS)	R/W to AS Rise Delay
13	TdDS (R/W)	DS Rise to R/W Valid Delay
14	TdDO (DSW)	Data Out to /DS (Write) Delay
15	ThDSW (DO)	/DS Rise (Write) to Data Out Hold Time
16	TdA (DI)	Address to Data In Required Valid Delay
17	TdAS (DSR)	/AS Rise to D/S (Read) Delay
18	TsDI (DSR)	Data in Setup Time to DS Rise (Read)
19	TdDM (AS)	/DM to /AS Rise Delay
20	TdDS (DM)	/DS Rise to /DM Valid Delay
21	ThDS (A)	/DS Rise to Address Valid Hold Time
22	TwW	Wait Width (One Wait) Window
23	TdAS (W)	/AS Rise to Wait Delay

The value of TsDI (DSR) has been measured for the NMOS part as mentioned below as TsDI (DSR) old. This "old" value needs to be relaxed as to the value described as TsDI (DSR) new. This new value will allow the customer to use external memories with slower access times that immediately translates in lower cost.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

AC ELECTRICAL CHARACTERISTICS

20 MHz Timing

No.	Symbol	Normal Min	Normal Max	Extended Min	Extended Max
1	TdA (AS)	25		70	
2	ThAS (A)	25		70	
3	TdAS (DI)		180		375
4	TwAS	35		85	
5	TdAZ (DSR)	0		0	
6	TwDSR	140		285	
7	TwDSW	85		185	
8	TdDSR (DI)		115		260
9	ThDSR (DI)	0		0	
10	TdDS (A)	25		25	
11	TdDS (AS)	20		65	
12	TdR/W (AS)	25		70	
13	TdDS (R/W)	20		65	
14	TdDO (DSW)	30		70	
15	ThDSW (DO)	20		65	
16	TdA (DI)		205		445
17	TdAS (DSR)	25		70	
18	TsDI (DSR)	25		25	
19	TdDM (AS)	20		65	
20	TdDS (DM)	20		65	
21	ThDS (A)	20		65	

Powered by ICminer.com Electronic-Library Service CopyRight 2003

AC ELECTRICAL CHARACTERISTICS
12 MHz Timing

No.	Symbol	Normal Min	Normal Max	Extended Min	Extended Max
1	TdA (AS)	55		135	
2	ThAS (A)	55		135	
3	TdAS (DI)		305		630
4	TwAS	70		150	
5	TdAZ (DSR)	0		0	
6	TwDSR	240		480	
7	TwDSW	150		320	
8	TdDSR (DI)		215		440
9	ThDSR (DI)	0		0	
10	TdDS (A)	55		130	
11	TdDS (AS)	45		125	
12	TdR/W (AS)	55		135	
13	TdDS (R/W)	45		125	
14	TdDO (DSW)	65		150	
15	ThDSW (DO)	45		125	
16	TdA (DI)		365		770
17	TdAS (DSR)	55		135	
18	TsDI (DSR)	25		25	
19	TdDM (AS)	50		130	
20	TdDS (DM)	45		125	
21	ThDS (A)	45		125	

Powered by ICminer.com Electronic-Library Service CopyRight 2003

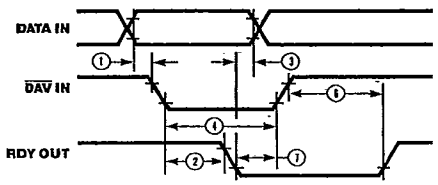
AC ELECTRICAL CHARACTERISTICS

25 MHz Timing

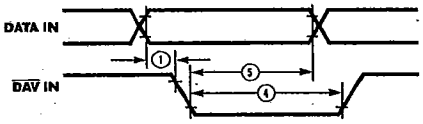
No.	Symbol	Normal Min	Normal Max	Extended Min	Extended Max
1	TdA (AS)	15		50	
2	ThAS (A)	15		50	
3	TdAS (DI)		140		280
4	TwAS	26		65	
5	TdAZ (DSR)	0		0	
6	TwDSR	110		220	
7	TwDSW	65		142	
8	TdDSR (DI)		85		195
9	ThDSR (DI)	0		0	
10	TdDS (A)	20		55	
11	TdDS (AS)	15		50	
12	TdR/W (AS)	15		50	
13	TdDS (R/W)	15		50	
14	TdDO (DSW)	20		50	
15	ThDSW (DO)	15		50	
16	TdA (DI)		155		330
17	TdAS (DSR)	15		50	
18	TsDI (DSR)	25		25	
19	TdDM (AS)	10		45	
20	TdDS (DM)	15		50	
21	ThDS (A)	15		50	

Powered by ICminer.com Electronic-Library Service CopyRight 2003

INPUT HANDSHAKE TIMING



Fully Interlocked Mode



Strobed Mode

AC ELECTRICAL CHARACTERISTICS (20 MHz)
Input Handshake

No.	Symbol	Parameter	Min	Max	Notes*†
1	TsDI(DAV)	Data In to Setup Time	0		
2	TdDAVIf(RDY)	/DAV Fall Input to RDY Fall Delay		200	1
3	ThDI(RDY)	Data in Hold Time from RDY Fall	0		
4	TwDAV	/DAV In Width	45		
5	ThDI(DAV)	Data In Hold Time from /DAV Fall	130		
6	TdDAV(RDY)	/DAV Rise Input to RDY Rise Delay		100	2
7	TdRDYf(DAV)	RDY Rise Output to /DAV Rise Delay	0		

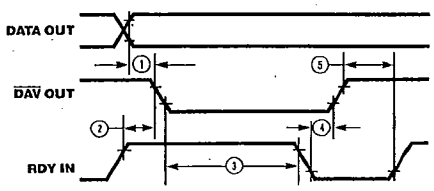
Notes:
1. Standard Test Load
2. This time assumes user program reads data before /DAV Input goes High. RDY will not go High before data is read.

* Times are given in nanoseconds.

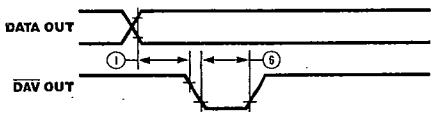
† Times are preliminary and subject to change.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

OUTPUT HANDSHAKE TIMING



Fully Interlocked Mode



Strobed Mode

AC ELECTRICAL CHARACTERISTICS (12 MHz, 20 MHz)
Output Handshake

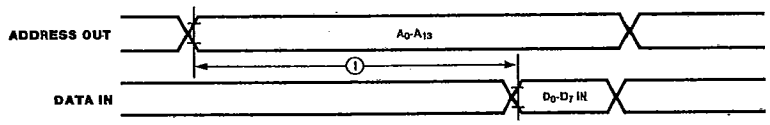
No.	Symbol	Parameter	Min	Max	Notes†*
1	TdDO(DAV)	Data Out to /DAV Fall Delay	90		1, 2
2	TdRDYr(DAV)	RDY Rise Input to /DAV Fall Delay	0	110	1
3	TdDAVOf(RDY)	/DAV Fall Output to RDY Fall Delay	0		
4	TdRDYf(DAV)	RDY Fall Input to /DAV Rise Delay	0	110	1
5	TdDAVOr(RDY)	/DAV Rise Output to RDY Rise Delay	0		
6	TwDAVO	/DAV Output Width	150		2

Notes:
1. Standard Test Load
2. Time given is for zero value in Deskew Counter. For non-zero value of n where n = 1,2, . . . 15 add 2 x n x TpC to the given time.

† Times given are in nanoseconds.
* Times are preliminary and subject to change.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

EPROM READ TIMING



EPROM READ Timing

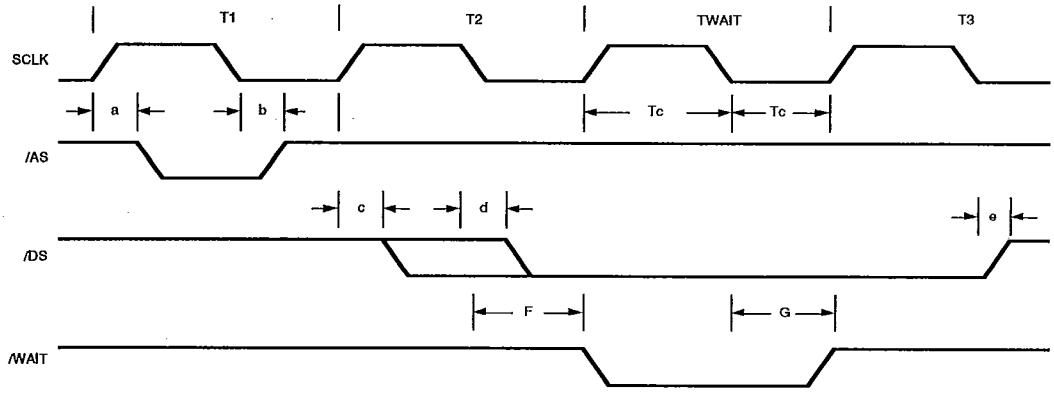
AC ELECTRICAL CHARACTERISTICS (20 MHz)
EPROM Read Cycle

No.	Symbol	Parameter	Min	Max	Notes†*
1	TdA(DR)	Address Valid to Read Data Required Valid		170	1

Notes:
1. WAIT states add 167 ns to these times.
† All times are in nanoseconds and are for 12 MHz input frequency.
* Timings are preliminary and subject to change.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

WAIT TIMING

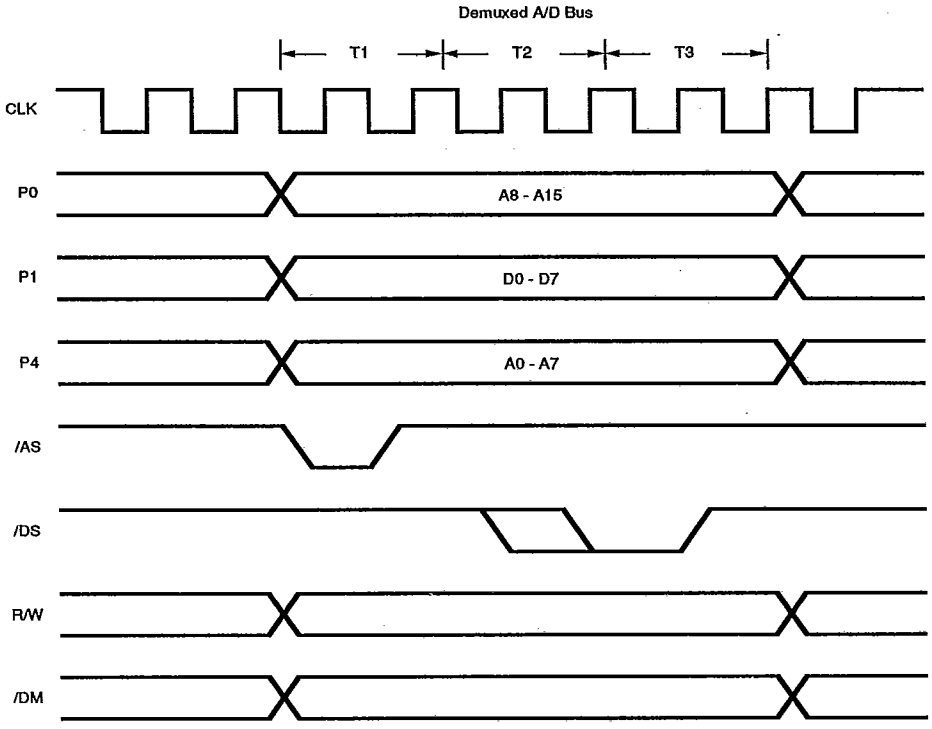


	Description		
a	Skew of T1 SCLK Rise to /AS Fall	10.0	Max
b	Skew of T1 SCLK Fall to /AS Rise	10.0	Max
c	Skew of T2 SCLK Rise to Read /DS Fall	20.0	Max
d	Skew T2 SCLK Fall to Write /DS Fall	20.0	Max
e	Skew T3 SCLK Fall to /DS Rise	20.0	Max
F	/WAIT Fall Delay After T2 SCLK Fall to Generate at Least 1 WAIT State	20.0	Max
G	/WAIT Rise Delay after TWAIT SCLK Fall to Prevent an Additional WAIT State	15.0	Max

Note:
All figures in nanoseconds.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

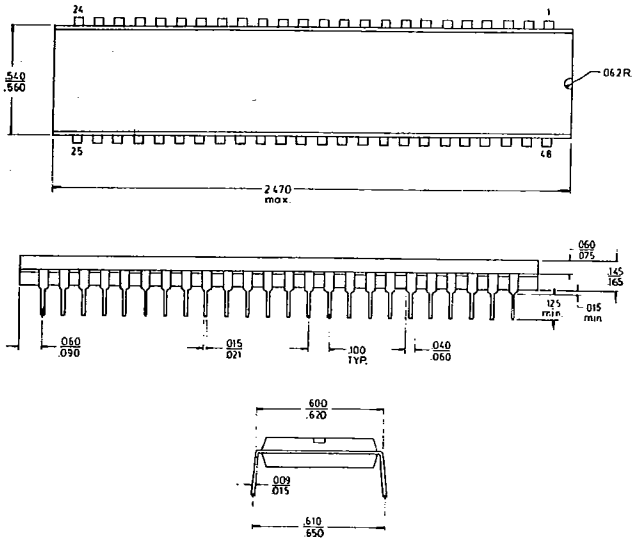
DE-MULTIPLEXED Z-BUS TIMING



Notes:
/AS, /DS, R/W, /DM Timing remains unchanged in demuxed A/D bus mode.

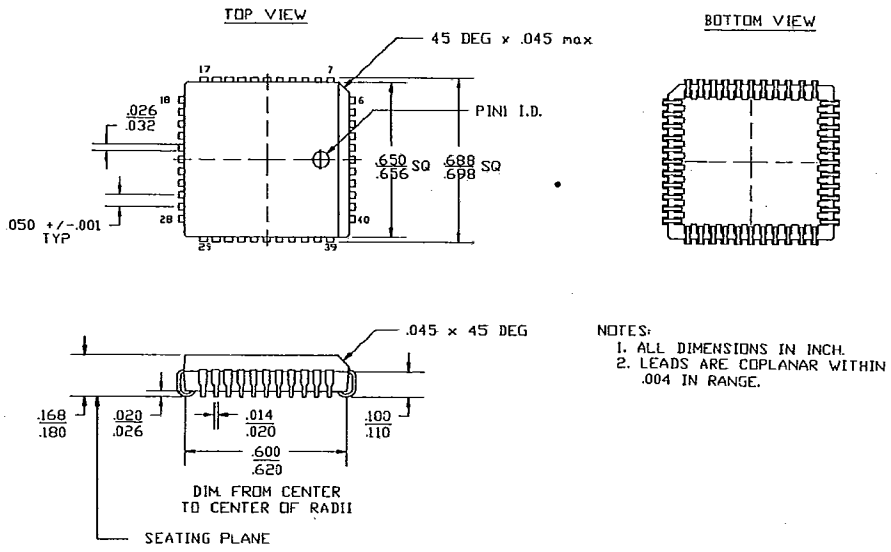
Powered by ICminer.com Electronic-Library Service CopyRight 2003

PACKAGE INFORMATION

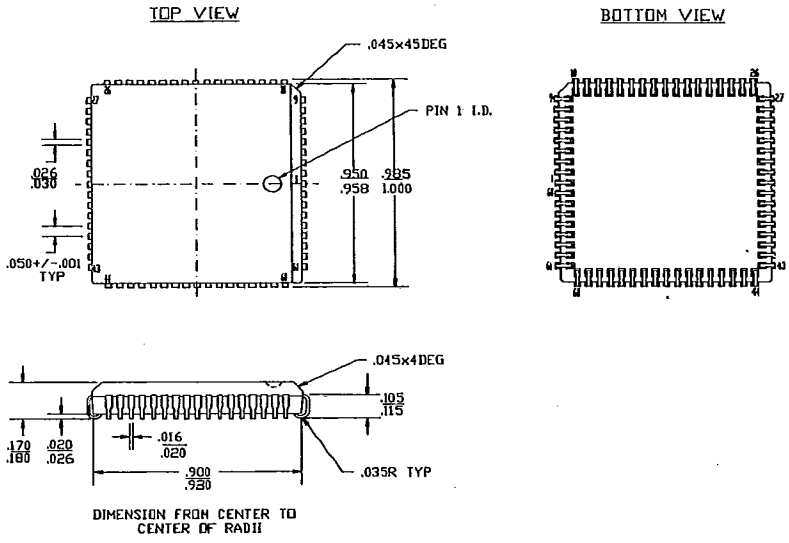


48-Lead DIP Package

Powered by ICminer.com Electronic-Library Service CopyRight 2003



44-Lead PLCC Package



68-Lead PLCC Package

Powered by ICminer.com Electronic-Library Service CopyRight 2003

ORDERING INFORMATION

Z88C00

20 MHz

68-Lead PLCC	48-Lead DIP	44-Lead PLCC
Z88C0020VSC	Z88C0020PSC	Z88C0120VSC
Z88C0020VEC	Z88C0020PEC	Z88C0120VEC

25 MHz

Z88C0025VSC	Z88C0025PSC	Z88C0125VSC
Z88C0025VEC	Z88C0025PEC	Z88C0125VEC

For fast results, contact your Zilog sales office for assistance in ordering the part desired.

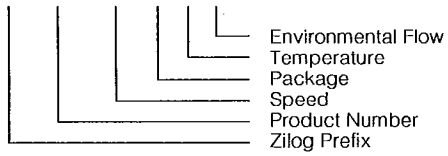
Package
P = Plastic DIP
V = Plastic Leaded Chip Carrier

Temperature
S = 0°C to +70°C
E = -40°C to +100°C

Speeds
20 = 20 MHz
25 = 25 MHz

Environmental
C = Plastic Standard

Example:
Z 88C00 20 V S C is an 88C00, 20 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



Powered by ICminer.com Electronic-Library Service CopyRight 2003