Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	Z8
Core Size	8-Bit
Speed	20MHz
Connectivity	UART/USART
Peripherals	DMA
Number of I/O	32
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z88c0020vsc

FEATURES

- Improved Z8® instruction set includes multiply and divide instructions, Boolean and BCD operations.
- Additional instructions support threaded-code languages, such as "Forth."
- 325 byte registers, including 272 general-purpose registers, and 53 mode and control registers.
- Addressing of up to 128K bytes of memory. Two register pointers allow use of short and fast instructions to access register groups within 600 nsec.
- Direct Memory Access controller (DMA).
- Two 16-bit counter/timers.
- Up to 32 bit-programmable and 8 byte-programmable I/O lines, with 2 handshake channels.
- Interrupt structure supports:
  - 27 interrupt sources
  - 16 interrupt vectors (2 reserved for future versions)
  - 8 interrupt levels
  - Servicing in 600 nsec. (1 level only)
- Full-duplex UART with special features.
- On-chip oscillator.
- 20 MHz clock.
- 8K byte ROM for Z8820

GENERAL DESCRIPTION

The Zilog Super8 single-chip MCU can be used for development and production. It can be used as I/O- or memory-intensive computers, or configured to address external memory while still supporting many I/O lines.

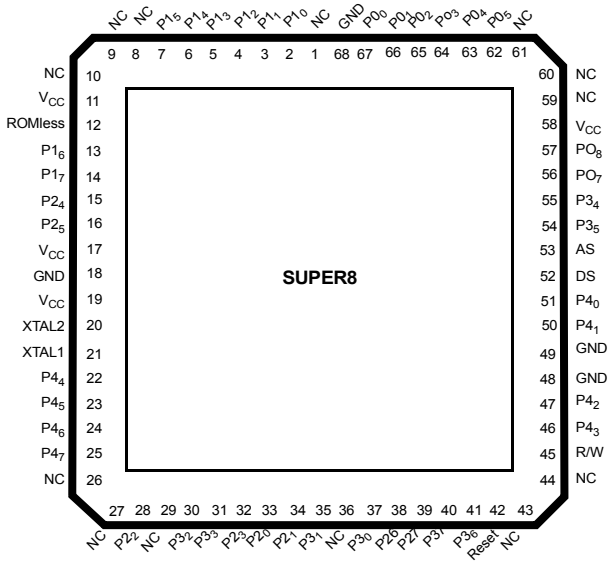

Figure 1.Pin Assignments 88-Pin PLCC

The Super8 features a full-duplex universal asynchronous receiver/ transmitter (UART) with on-chip baud rate generator, two programmable counter/timers, a direct memory access (DMA) controller, and an on-chip oscillator.

The Super8 is also available as a 48-pin and 68-pin ROMless microcomputer with four byte-wide I/O ports plus a byte-wide address/data bus. Additional address bits can be configured, up to a total of 16.

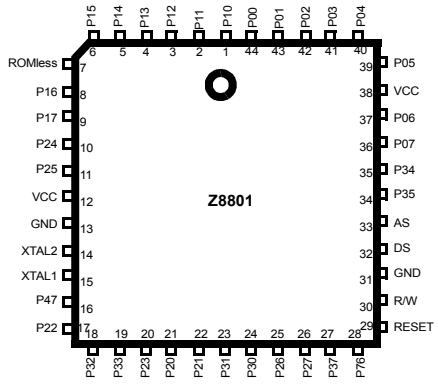

Figure 2.Pin Assignments 44-Pin PLCC

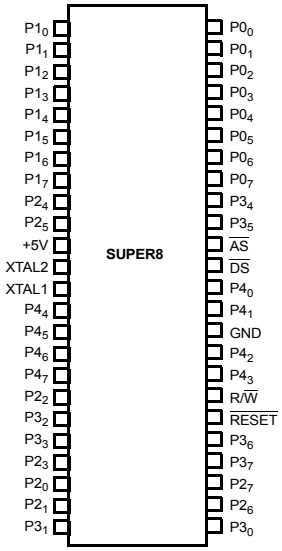

Figure 3.Pin Assignments 48-Pin DIP

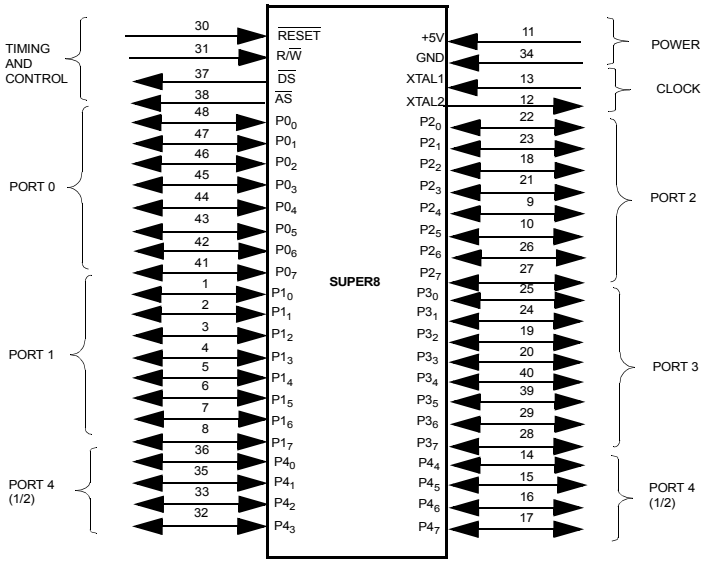

Figure 4.Pin Functions 48-Pin DIP

oped, it can be mask-programmed into the production microcomputer device, directly replacing the emulator. The protopack part is also useful in situations where the cost of mask-programming is prohibitive or where program flexibility is desired.

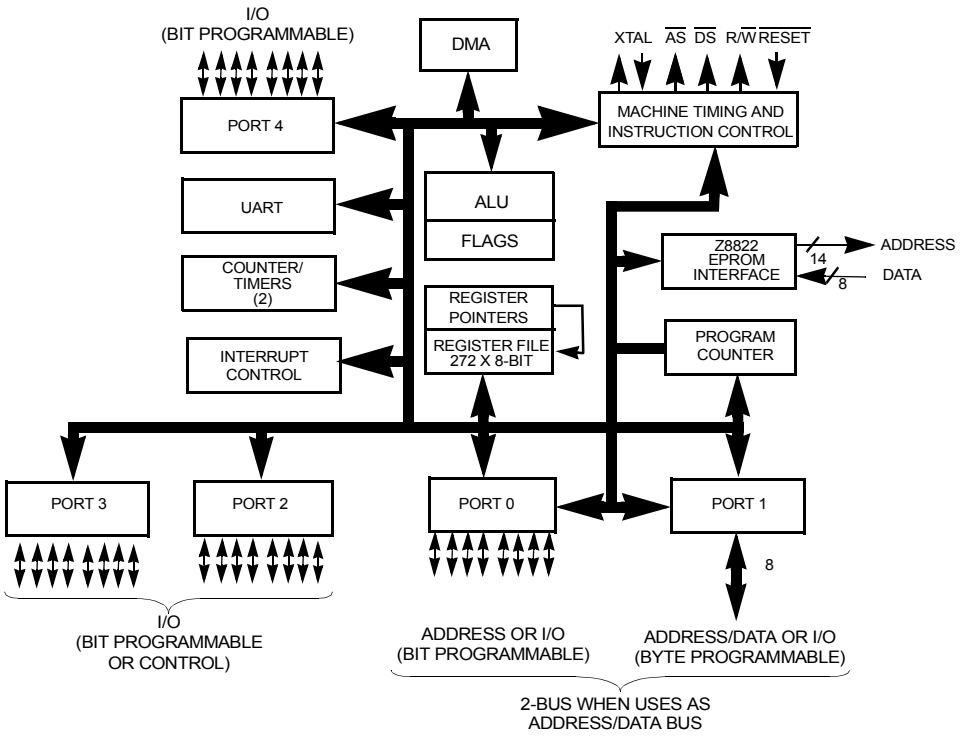

Figure 7.Functional Block Diagram

ARCHITECTURE

The Super8 architecture includes 325 byte-wide internal registers. 272 of these are available for general purpose use; the remaining 53 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.

ter pointer remains unchanged, the three bits from the address 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 RP0 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 (CO 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 9.

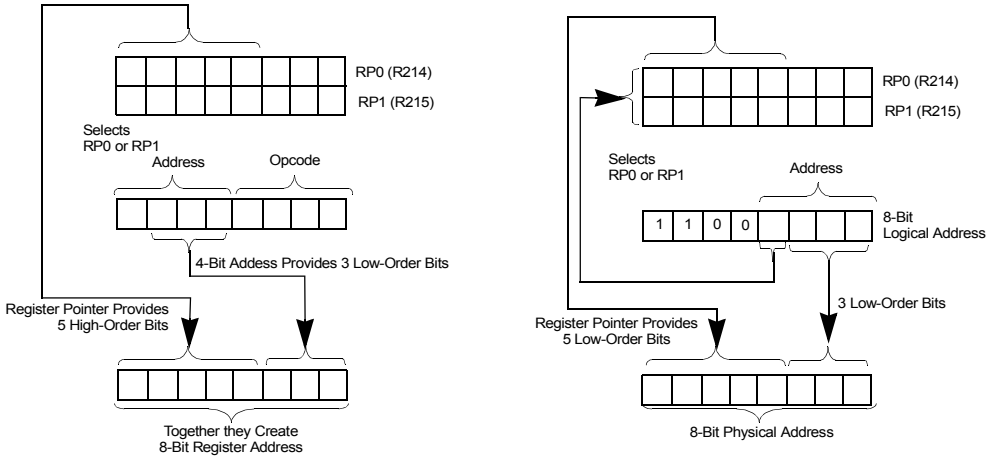

Figure 9.Working Register Window

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

Register List

Super-8 Registers lists the Super8 registers. For more details, see Figure 10.

Table 15.Super-8 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	DI		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 High Byte
217	D9		SPL	Stack Pointer Low 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
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	CTCH	CTR 0 Timer Constant, bits 8-15
227	E3	Bank 0	C0CL	CTR 0 Capture Register, bits 0-7

Table 15.Super-8 Registers (Continued)

Address				
Decimal	**Hexadecimal**		**Mnemonic**	**Function**
		Bank 1	CTCL	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
235	EB	Bank 0	UTC	UART Transmit Control
236	EC	Bank 0	URC	UART Receive Control
237	ED	Bank 0	UIE	UART Interrupt Enable
239	EF	Bank 0	UIO	UART Data
240	F0	Bank 0	POM	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	HOC	Handshake Channel 0 Control
245	F5	Bank 0	H1C	Handshake Channel I Control
246	F6	Bank 0	P4D	Port 4 Direction
247	F7	Bank 0	P40D	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

Table 15.Super-8 Registers (Continued)

Address				
Decimal	**Hexadecimal**		**Mnemonic**	**Function**
		Bank 1	WUMCH	Wakeup Match Register
255	FF	Bank 0	IPR	Interrupt Priority Register
		Bank 1	WUMSK	Wakeup Match Register

MODE AND CONTROL REGISTERS

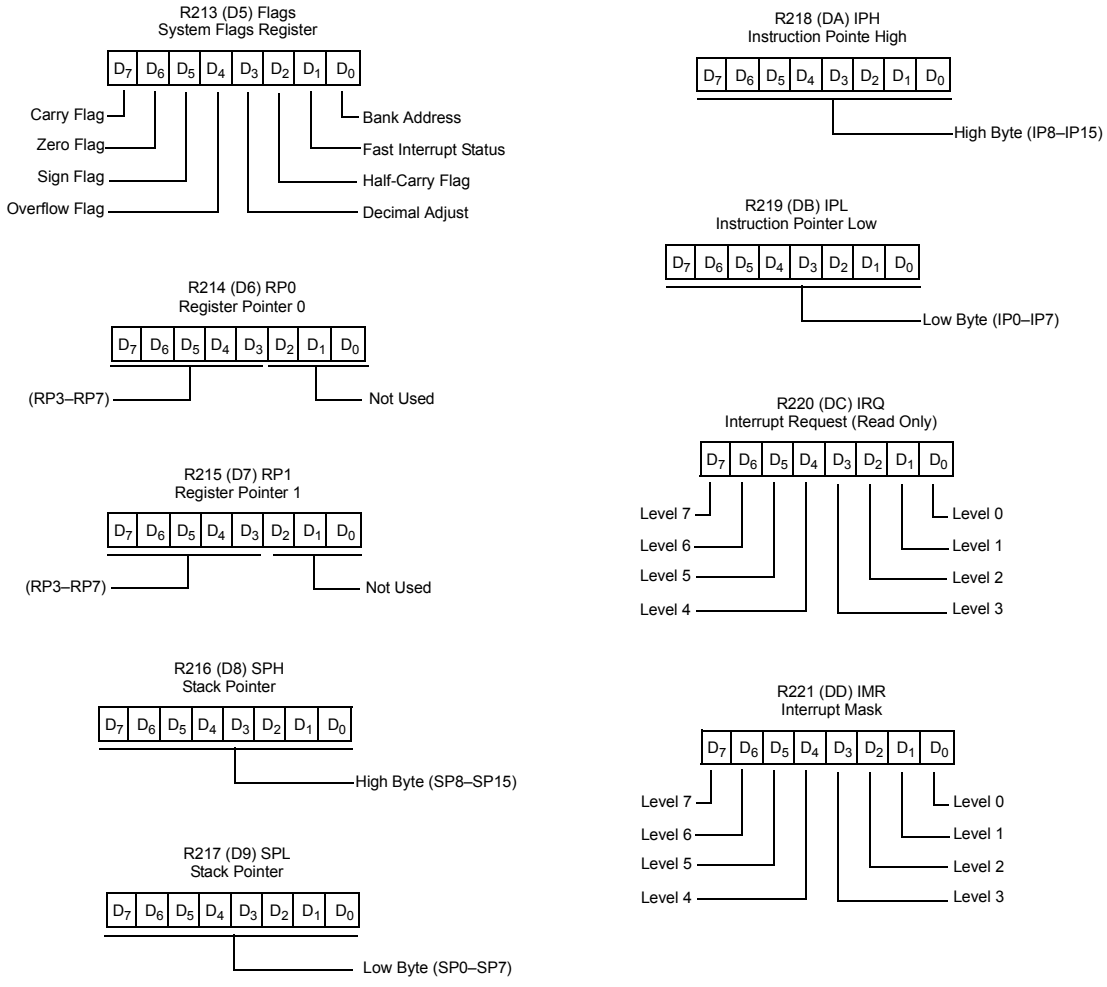

Figure 10.Mode and Control Registers

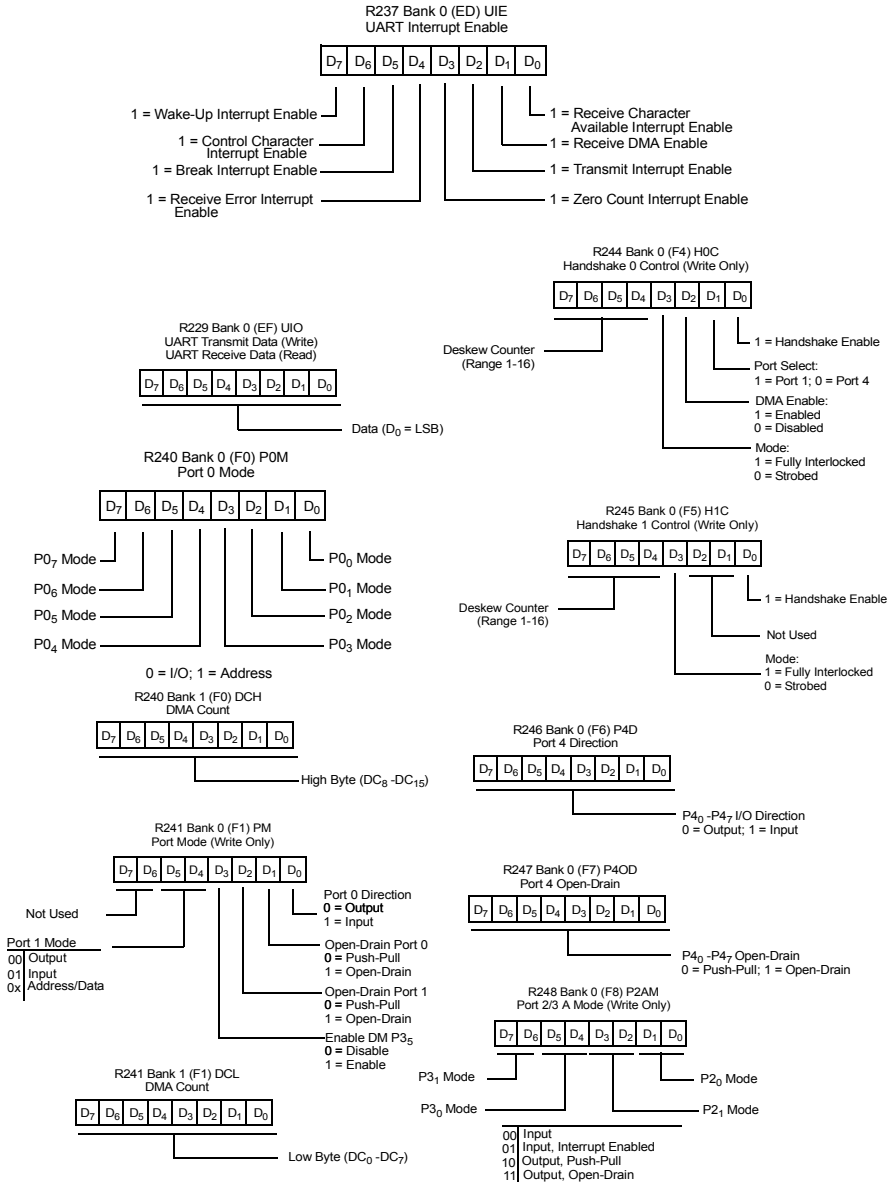

Figure 13. Mode and Control Registers (Continued)

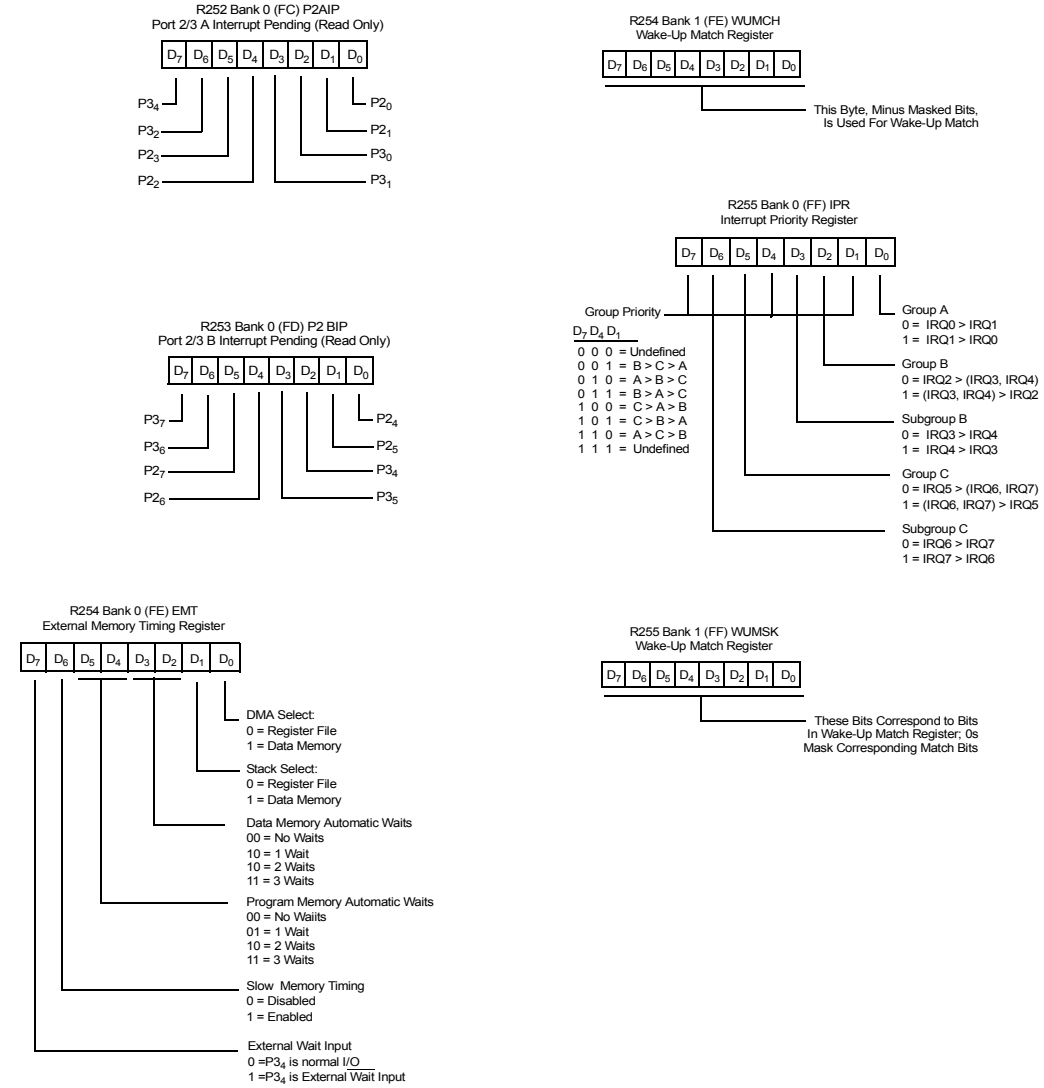

Figure 15.Mode and Control Registers (Continued)

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 inputs or outputs. Some can also be configured as address/data lines.

Port 1

In the ROMless device, Port 1 is configured as a byte-wide address/data port. It provides a byte-wide multiplexed address/data path. Additional address lines can be added by configuring Port 0.

The ROM and Protopack Port 1 can be configured as above or as an I/O port; it can be a byte-wide input, open-drain output, or push-pull output. It can be placed under handshake control or handshake channel 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 3.

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 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 17.Pin Assignments for Ports 2 and 3

Port 2		Port 3	
Bit	**Function**	**Bit**	**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/$\overline{\text{WAIT}}$
5	Handshake 0 output	5	Handshake 1 output/$\overline{\text{DM}}$
6	Counter 0 input	6	Counter 1 input
7	Counter 0 I/O	7	Counter 1 I/O

controlled by the $\overline{DM}$ line (Port $P3_5$), which selects data memory when Low and program memory when High.

Figure 16 on page 23 shows the system memory space.

CPU Program Memory

Program memory occupies addresses 0 to 64K. External program memory, if present, is accessed by configuring Ports 0 and 1 as a memory interface.

The address/data lines are controlled by $\overline{AS}$, $\overline{DS}$ and R/$\overline{W}$.

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.

ROMless

Port 0 can be configured to provide from 0 to 8 additional address lines. Port 1 is always used as an 8-bit multiplexed address/data port.

ROM and Protopack

Port 1 is configured as multiplexed address/data or as I/O. When Port 1 is configured as address/data, Port 0 lines can be used as additional address lines, up to address 15. External program memory is mapped above internal program memory; that is, external program memory can occupy any space beginning at the top of the internal ROM space up to the 64K (16-bit address) limit.

CPU Data Memory

The external CPU data memory space, if separated from program memory by the $\overline{DM}$ optional output, can be mapped anywhere from 0 to 64K (full 16-bit address space). Data memory uses the same address/data bus (Port 1) and additional addresses (chosen from Port 0) as program memory. Data memory is distinguished from program memory by the DM pin ($P3_5$), and by the fact that data memory can begin at address OOOOH. This feature differs from the Z8.

Table 20.Instruction Summary (Continued)

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
INCW dst	RR		A0	-	*	*	*	-	-
dst←1 + dst	IR		A1						
IRET (Fast)			BF	Restored to before interrupt					
PC↔IP									
FLAG←FLAG‘									
FIS←0									
IRET (Normal)			BF	Restored to before interrupt					
FLAGS←@SP; SP←SP + 1									
PC←@SP; SP←SP + 2; SMR(0)←1									
JP cc, dst	DA		ccD	-	-	-	-	-	-
if cc is true,			(cc = 0 to F)						
PC←dst	IRR		30						
JR cc, dst	RA		ccB	-	-	-	-	-	-
if cc is true,									
PC←PC + d			(cc = 0 to F)						
LD dst, src	r	IM	rC	-	-	-	-	-	-
dst←src	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	F5						
	r	x	87						
	x	r	97						

Table 20.Instruction Summary (Continued)

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
PC←@SP; SP←SP + 2									
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	*	*	*	*	-	-
SB0 BANK←0			4F	-	-	-	-	-	-
SB1 BANK←1			5F	-	-	-	-	-	-
SBC dst, src dst←dst – src – C	Note[1]		3[ ]	*	*	*	*	1	*
SCF			DF	1	-	-	-	-	-

Table 21.Second Nibble

Addr Mode		Lower Opcode Nibble[1]
dst	src	
r	r	[2]
r	Ir	[3]
R	R	[4]
R	IR	[5]
R	IM	[6]

1. For example, to use an opcode represented as x[ ] with an "RR" addressing mode, use the opcode "x4."
   0= Cleared to Zero
   1= Set to One
   -= Unaffected
   *= Set or reset, depending on result of operation.
   U= Undefined

INSTRUCTIONS

Table 22.Super8 Instructions

Mnemonic	Operands	Instruction
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 and 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

- 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-triggered flip-flops to convert the edges to levels.

The level-activated system requires that interrupt-serving 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.

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 top 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 are 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 grows from low

addresses to high addresses in the register file, use PUSHUI and POPUD. For a stack that grows from high addresses 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 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 cause 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 7.

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 take ten CPU clock cycles, excluding wait states.

OUTPUT HANDSHAKE TIMING

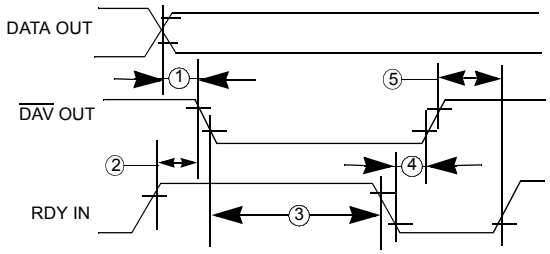

Figure 24.Fully Interlocked Mode

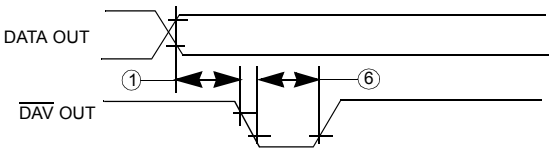

Figure 25.Strobed Mode

AC CHARACTERISTICS (12 MHz, 20 MHz)

Output Handshake

Table 25.AC Characteristics (12 MHz, 20 MHz) Output Handshake

Number	Symbol	Parameter	Min	Max	Notes[1,2]
1	TdDO(DAV)	Data Out to $\overline{DAV}$ ↓ Delay	90		Note[3,4]
2	TdRDYr(DAV)	RDY ↑ Input to $\overline{DAV}$ ↓ Delay	0	110	Note[3]
3	TdDAVOf(RDY)	$\overline{DAV}$ ↓ Output to RDY ↓ Delay	0		
4	TdRDYf(DAV)	RDY ↓ Input to $\overline{DAV}$ ↑ Delay	0	110	Note[3]
5	TdDAVOr(RDY)	$\overline{DAV}$ ↑ Output to RDY ↑ Delay	0		
6	TwDAVO	$\overline{DAV}$ Output Width	150		Note[4]

1. Times are preliminary and subject to change.
2. Times given are in ns.
3. Standard Test Load
4. Time given is for zero value in Deskew Counter. For nonzero value of n where n = 1, 2,. . . 15 add 2 x n x TpC to the given time.

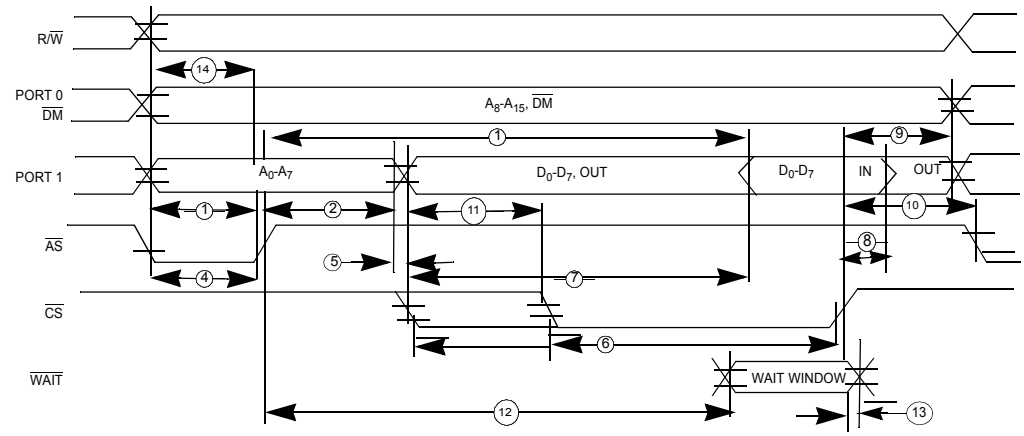

Figure 26.External Memory Read and Write Timing

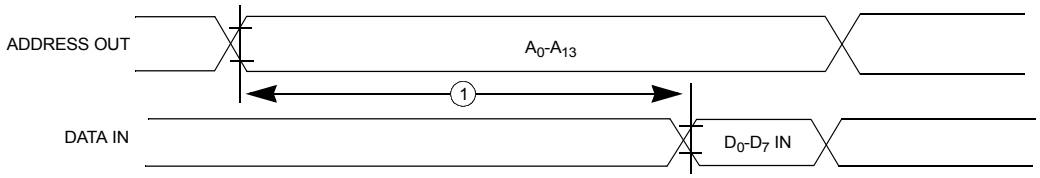

Figure 27.EPROM Read Timing