OPEN ELECTIVE EC604(A)
MICROCONTROLLERS AND EMBEDDED SYSTEM
UNIT 2

By Dr Sangeeta Shukla SIRT, Bhopal

16 Bit Microcontrollers 8096

MCS-96 family members are all high-performance microcontrollers with a 16-bit CPU and at least 230 bytes of on-chip RAM.

They are register-to-register machines, so no accumulator is needed, and most operations can be quickly performed from or to any of the registers.

In addition, the register operations can control the many peripherals which are available on the chips. These peripherals include a serial port, A/D converter, PWM output, up to 48 I/O lines and a High- Speed I/O subsystem which has 2 16-bit timer/counters, an 8-level input capture FIFO and an 8-entry programmable output generator.

MCS-96 products are being used in modems, motor controls, printers, engine controls, photocopiers, antilock brakes, air conditioner temperature controls, disk drives, and medical instrumentation.

The 80C196KB can be separated into four sections for the purpose of describing its operation.

Functional Block Diagram of 8096 Microcontrollers

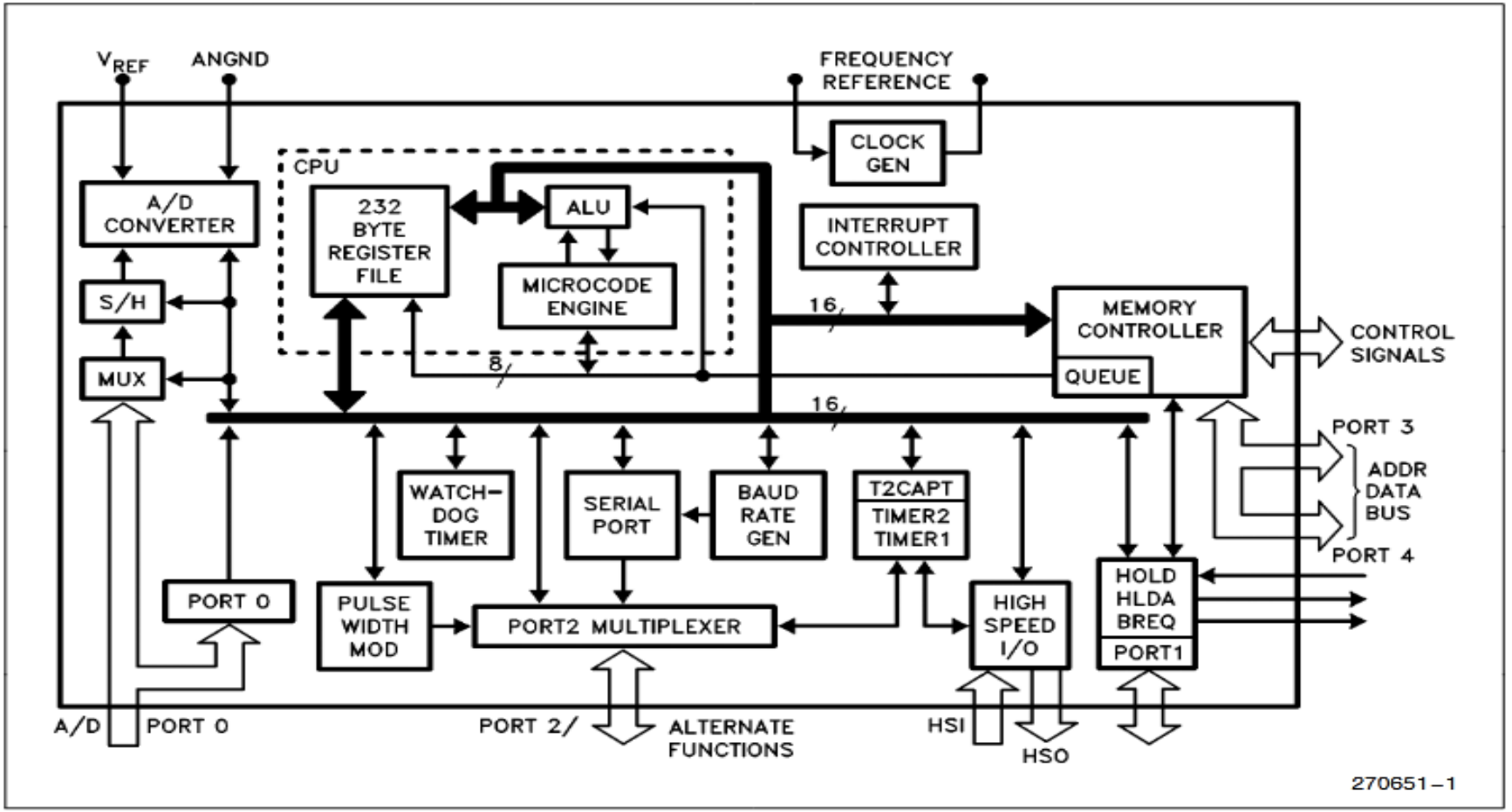


Figure 1-1. 80C196KB Block Diagram

Following are the resources offered by the 8096 microcontroller.

1. 16-bit CPU
2. On-chip clock generator
3. 64 KB memory space
4. 256 bytes on-chip RAM
5. 8 KB on-chip ROM (*)
6. 26 special function registers
7. 4/8 analog input channels (*)
8. 20/24/40 digital I/O lines (*)
9. 4 high speed inputs
10. 6 high speed outputs
11. 2–16 bits timers/counters
12. 7/8 source interrupt structure
13. 1 D/A output channel
14. Full duplex serial I/O port
15. Watchdog timer

(*) The resources like analog inputs and on-chip ROM are available oncertain versions only.

The MCS-96 family of microcontrollers

Micro-controller	*Pins*	*CPU*	*On-chip RAM*	*On-chip ROM*	*Analog channels (*)*	*Digital I/O lines (*)*	*High speed input (**)*	*High speed output (**)*	*Timer counter*	*Interrupt source*	*PWM (D/A output channel)*	*Serial I/O ports*	*Watch-dog timer*
8096	68	16 bits	256 bytes	Nil	Nil	40	4	6	2	7	1	1	Yes
8094	48	16 bits	256 bytes	Nil	Nil	24	4	6	2	7	1	1	Yes
8396	68	16 bits	256 bytes	8 KB	Nil	40	4	6	2	7	1	1	Yes
8394	48	16 bits	256 bytes	8 KB	Nil	24	4	6	2	7	1	1	Yes
8097	68	16 bits	256 bytes	Nil	8	24	4	6	2	8	1	1	Yes
8095	48	16 bits	256 bytes	Nil	4	20	4	6	2	8	1	1	Yes
8397	68	16 bits	256 bytes	8 KB	8	24	4	6	2	8	1	1	Yes
8395	48	16 bits	256 bytes	8 KB	4	20	4	6	2	8	1	1	Yes

Note:(*) There are five ports P0, P1, P2, P3 and P4 each of 8 bits.

Port 0 is shared with eight analog channels both in case of the 8097 and the 8397.

4 pins of Port 0 are shared with four analog channels both in case of the 8095 and the 8395.

Following are not provided on the 48-pin package, i.e. in case ofthe 8094, the 8394, the 8095 and the 8395.

(a) 8 pins of Port 1
(b) 4 pins of Port 0 (P0.0–P0.3)
(c) 4 pins of Port 2 (P2.3, P2.4, P2.6, P2.7)

Port 3 and port 4 are shared with address/data lines.
They can be used for interface to external memory and/or I/O.

MEMORY ORGANIZATION

The 8096 can access up to 64 KB of memory

The Scratch Pad Registers(called Register File), Special Function Registers, On-chip RAM, On-chip ROM (in case of the 8396, the 8394, the 8397 and the 8395) and External Memory Space are the main constituents of memory.

MEMORY ORGANIZATION

The basic blocks are as follows:

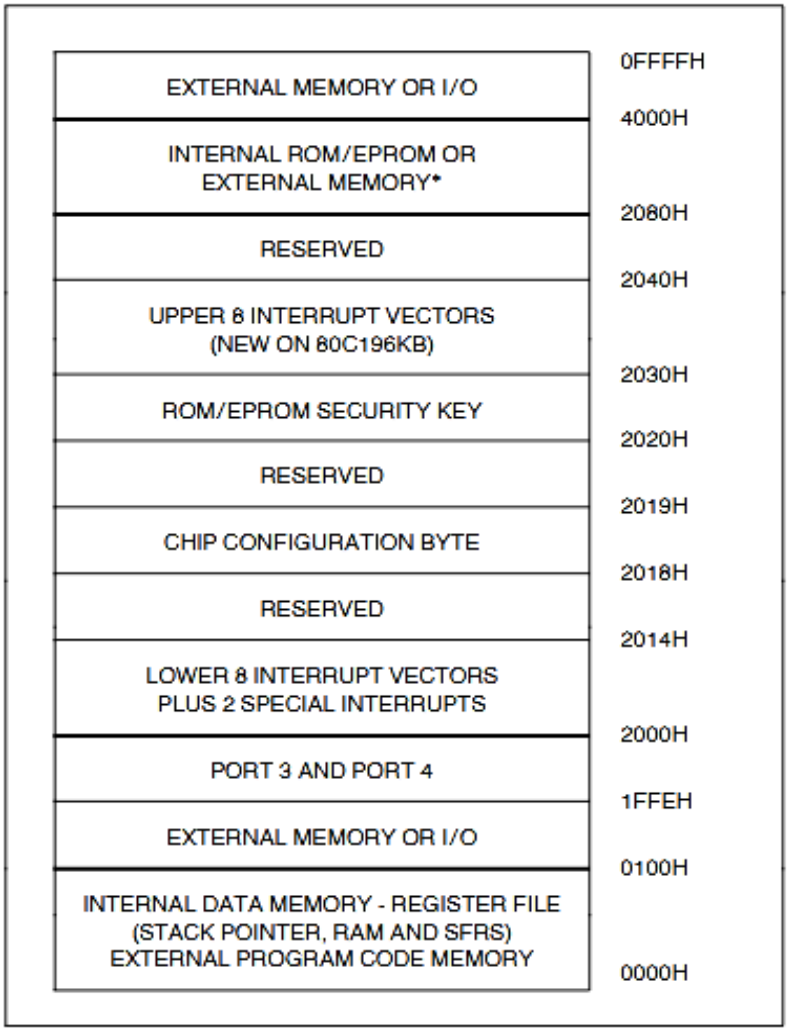
(i) Internal RAM containing

- Special function registers (00H to 17H)
- Stack pointer (18H and 19H)
- Register file (1AH to EFH
- Power down RAM (F0H to FFH)

This memory area is accessed as data memory and no code may be executed from this area.

The program memory area of 00 to FFH is reserved for internal use of Intel development systems.

Locations 00H through 0FFH contain the Register Fileand Special Function Registers,



Memory Map

MEMORY ORGANIZATION

The basic blocks are as follows:

(ii) Internal ROM:

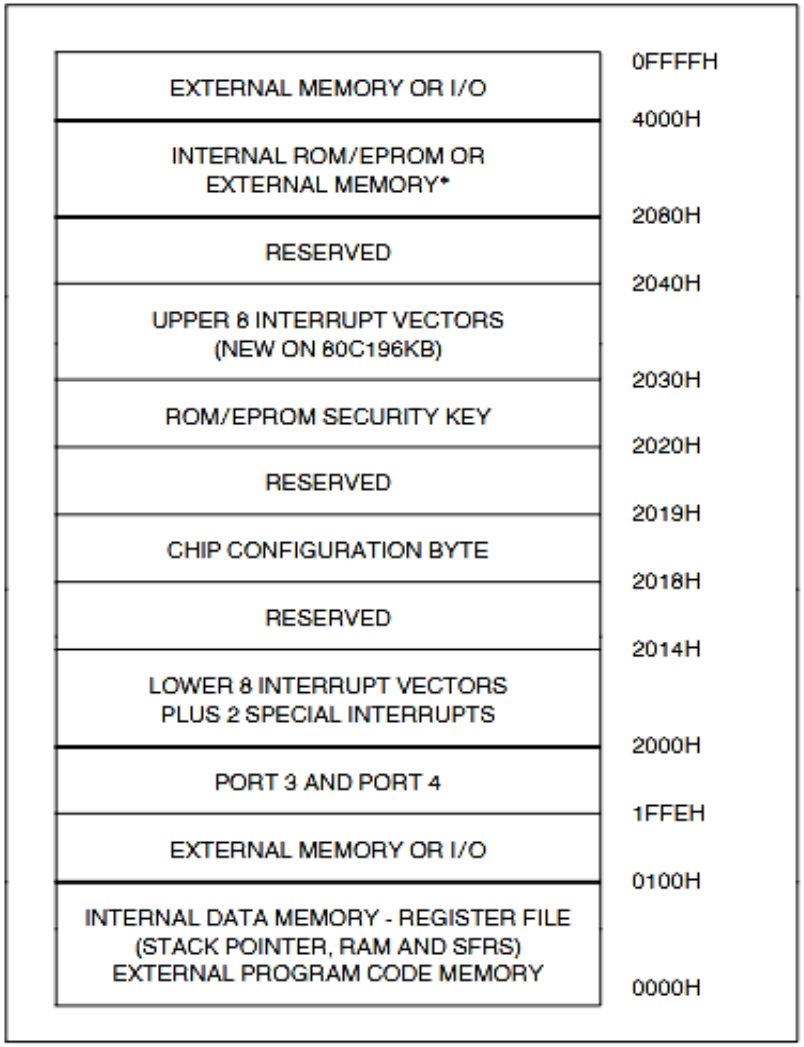
In case the chip has on-chip ROM, it would contain

- Interrupt vectors: An interrupt vector is the memory location of an interrupt handler, which prioritizes interrupts and saves them in a queue if more than one interrupt is waiting to be handled.

- Factory test code

Internal program storage and be available at addresses (2000H–3FFFH).

If the chip does not contain ROM, then



Memory Map

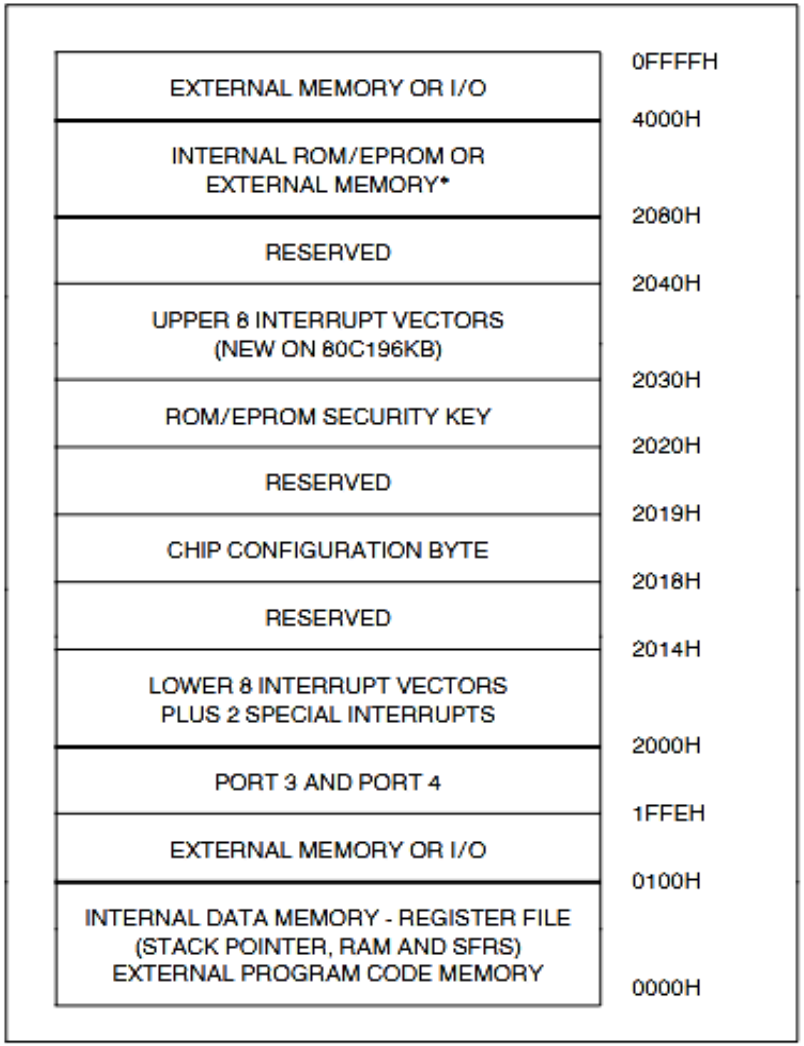
MEMORY ORGANIZATION

The basic blocks are as follows:

(iii) External memory or I/O available at addresses (0100H–1FFDH) and (4000H–FFFFH)

(iv) Ports 3 and 4 locations (1FFEH and 1FFFH) for reconfiguration,if these are not used as address/data lines.

When the 8096 is reset, address 2080H is loaded to the Program Counter to give 8 KB of continuous memory.

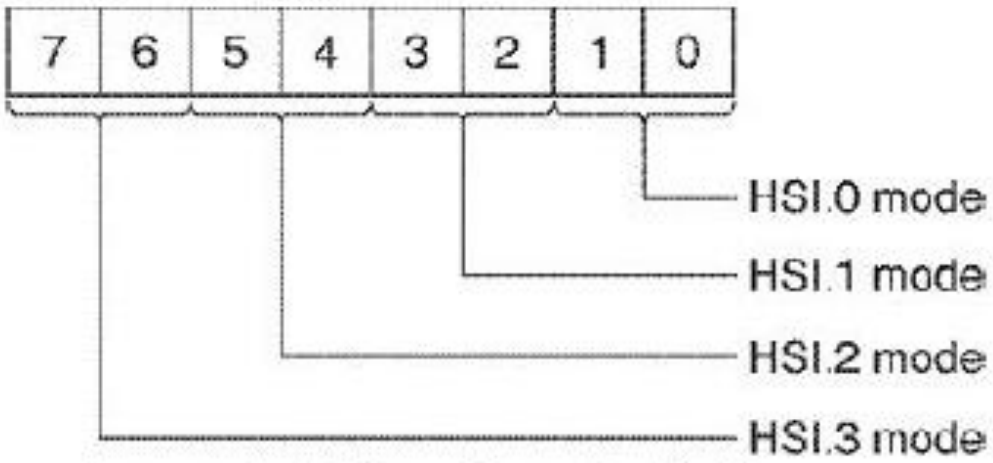


Memory Map

HIGH SPEED INPUT UNIT

- The 8096 provides four High Speed Input (HSI) lines.
- A total of eight events can be recorded.
- HSI.2 and HSI.3 share pins with High Speed Output pins 4 and 5.
- The function of these pins (HSI.2/HSO.4 and HSI.3/HSO.5) is controlled by I/O Control Register 0 (bits 4 and 6) and I/O Control Register 1 (bits 4 and 6).
- There are four possible modes of each High Speed Input line.
- The modes basically signify the type of signal events that will be recorded.

HIS_MODE REGISTER

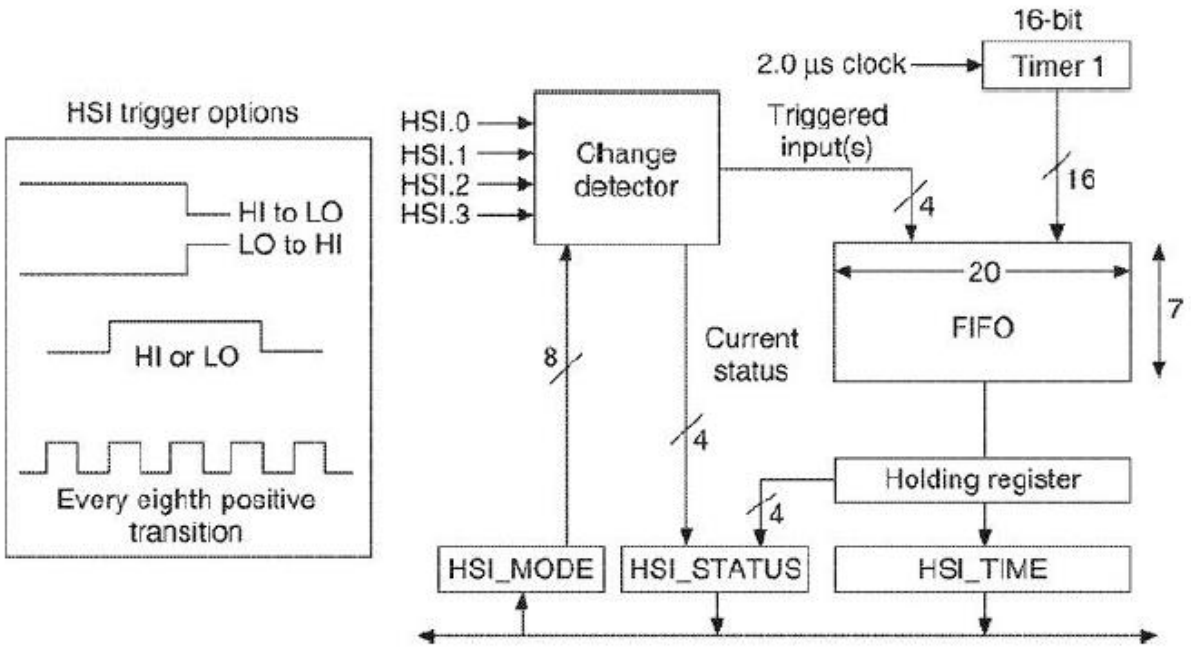


where each 2-bit mode control field defines one of the four possible modes:

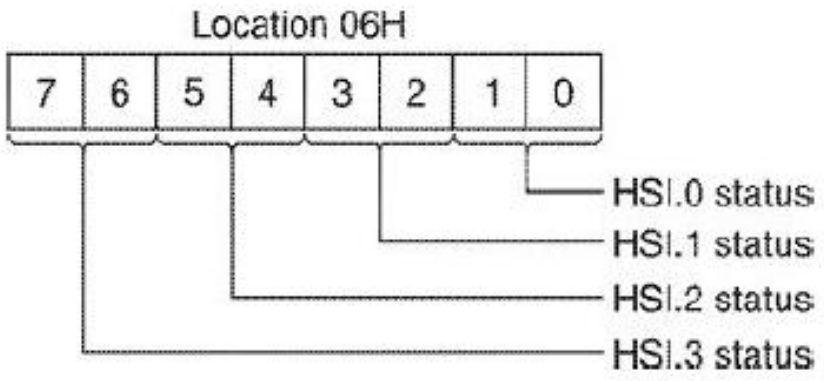
00 Eight positive transitions
01 Each positive transition
10 Each negative transition
11 Every transition (positive and negative)

HIGH SPEED INPUT UNIT

- Depending on the modes set,
- The change detector circuit generates event (triggered input) which is stored in FIFO along with the Timer 1 contents.
- The FIFO is 20-bit wide with 4 bits meant for status of four HSI lines and 16 bits for Timer 1 contents.
- Seven entries (seven events) in total can be stored in FIFO.
- The holding register is used to store one additional event when FIFO is full.



HIS_STATUS REGISTER



Shows the current status of HSI pins.

- Two bits of the register are dedicated to each HSI pin.
- The lower bit indicates whether or not the event has occurred on the pin at the HSI_TIME.
- The upper bit indicates the current status of the pin.
- The status of FIFO is indicated in I/O status register 1 (bits 6 and 7).

HIGH SPEED OUTPUT UNIT

The 8096 can be programmed to trigger different events at pre-specified time. The events are:

Starting of A/D conversion

Resetting of Timer 2

Generation of two interrupts

Setting software timers

Switching up to six output lines.

The six output lines of High Speed Output (HSO) unit are HSO.0–HSO.5.

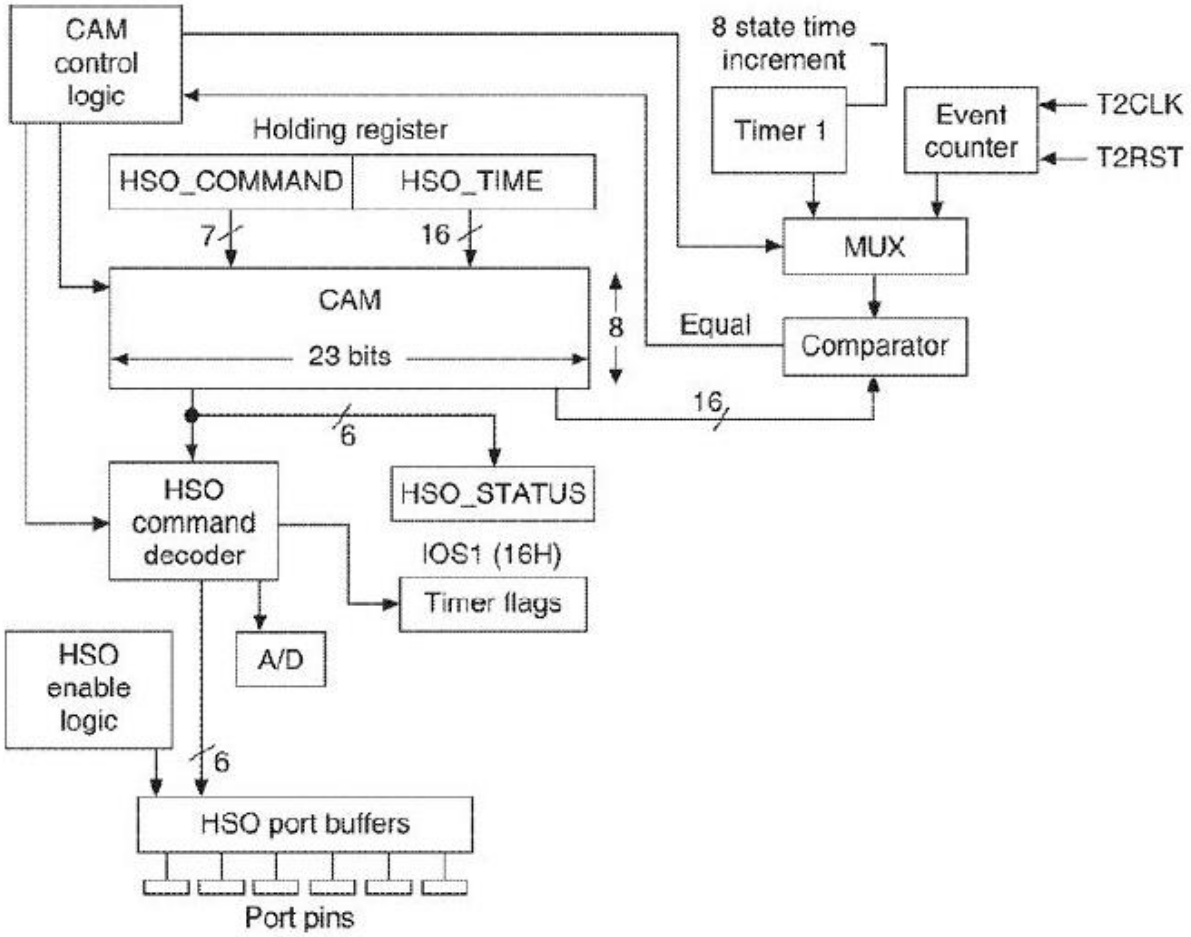
HSO.4 and HSO.5 are shared with HSI.2 and HSI.3 and this function is controlled by I/O Control Register1.

HIGH SPEED OUTPUT UNIT

The HSO unit contains a Content Addressable Memory (CAM) in which the time of the event (16 bits) and HSO command (7 bits) to trigger the event are stored.

These are stored through HSO_COMMAND and HSO_TIME registers which are special function registers.

A total of eight entries can be stored in CAM and when it is full, then one entry can be stored in the Holding register.



HIGH SPEED OUTPUT UNIT

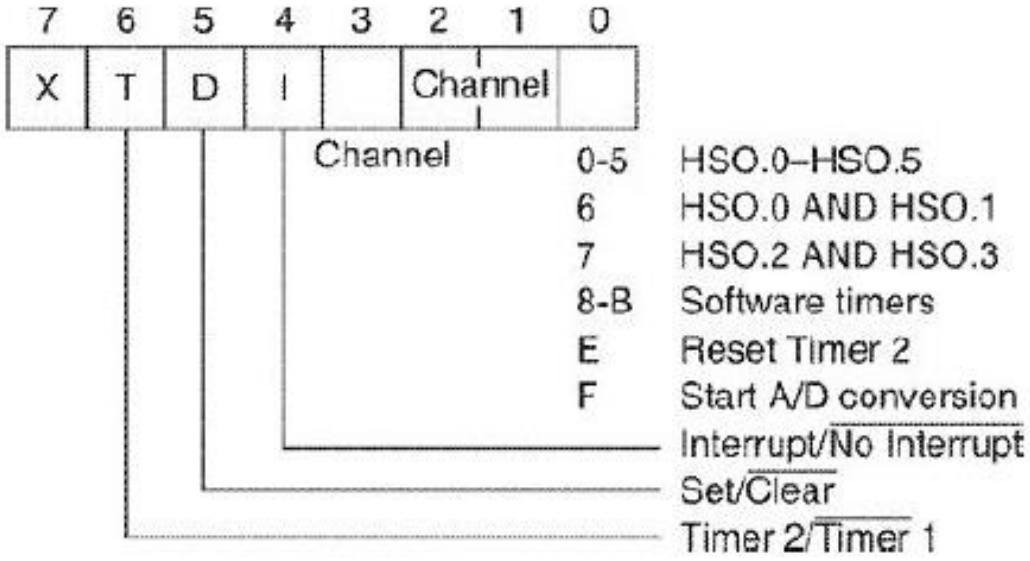
- All these events are triggered at pre-specified time.
- It takes one state time to check one location of CAM. Thus, eight state times are required to check all the locations in CAM. Hence, the minimum time difference between two events will be eight state times.
- Timer 2 in HSO has also been synchronized to change once per eight state times.
- The I/O status register 0 specifies status of the CAM through bits 6 and 7. If

  Bit 7 = 0: Holding register is empty.

  Bit 6 = 0: Holding register and at least one entry in CAM is empty.
- Depending on the status, further entries should be stored in CAM.

HSO COMMAND TAG FORMAT

- HSO command specifies (bit 6) whether Timer 1 or Timer 2 (Event Counter) should be taken for time comparison.
- The bit 5 is ignored.



8096 INPUT/OUTPUT PORTS

The 8096 contains Five 8-bit I/O ports which are shared with otherfunctions as well.
The ports are buffered both at input and output.

Port 0: It is an 8-bit high impendence, input only port. Its pins can be used as digital inputs and/or analog inputs to the on-chip A/D converter.These pins are also mode input to EPROM parts in the programming mode.

Port 1: It is a quasi-bidirectional port and can be used for sending/receiving 8-bit parallel data.

Port 2: It is a multifunction port since six of its pins are shared withother functions of the 8096 as shown in Table

8096 INPUT/OUTPUT PORTS

Port	*Function*	*Alternate function*	*Controlled by*
P2.0	output	TXD (Serial port transmit)	IOC1.5
P2.1	input	RXD (Serial port receive)	N/A
P2.2	input	EXTINT (External interrupt)	IOC1.1
P2.3	input	T2CLK (Timer 2 input)	IOC0.7
P2.4	input	T2RST (Timer 2 reset)	IOC0.3
P2.5	output	PWM (Pulse width modulation)	IOC1.0
P2.6	quasi-bidirectional		
P2.7	quasi-bidirectional		

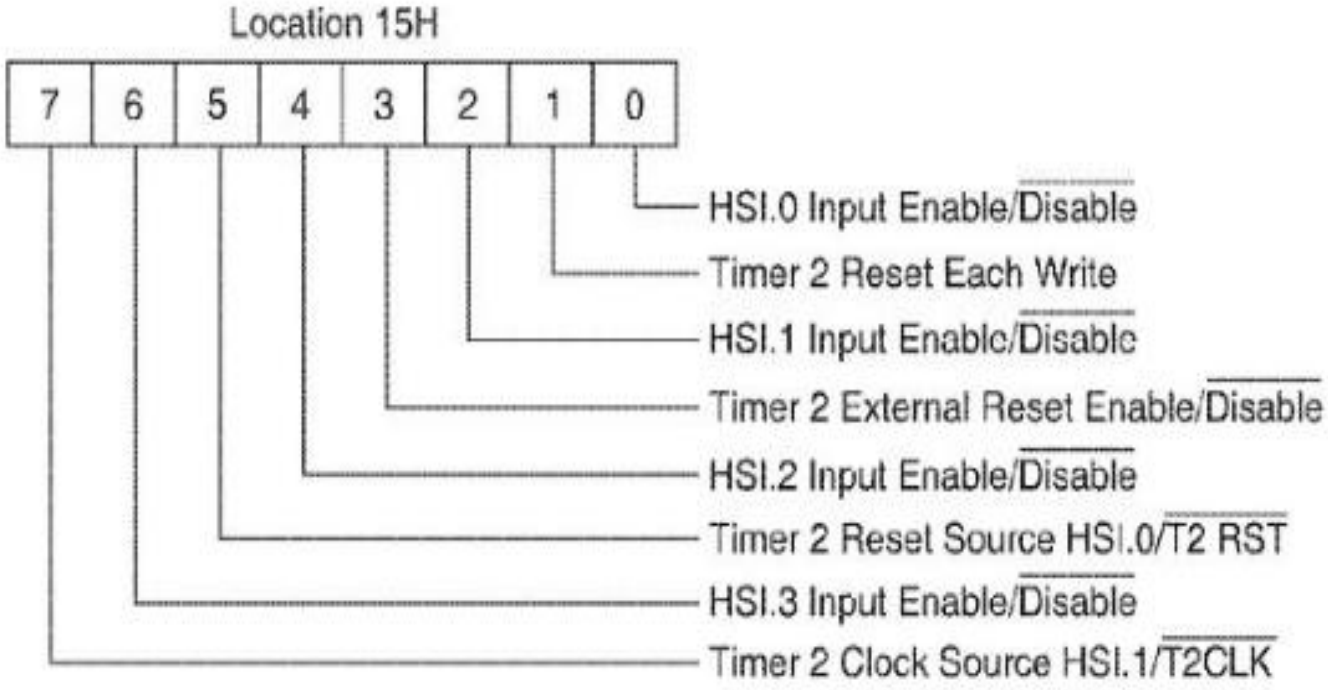
8096 INPUT/OUTPUT PORTS

Port 3 and Port 4: Port 3 and Port 4 can be used as bidirectional ports ifthe external memory is not required. In case of the external memory,these are used for the multiplexed address and data lines (P3.0–AD0,P3.7–AD7; P4.0–AD8, P4.7–AD15).

8096 I/O CONTROL REGISTERS

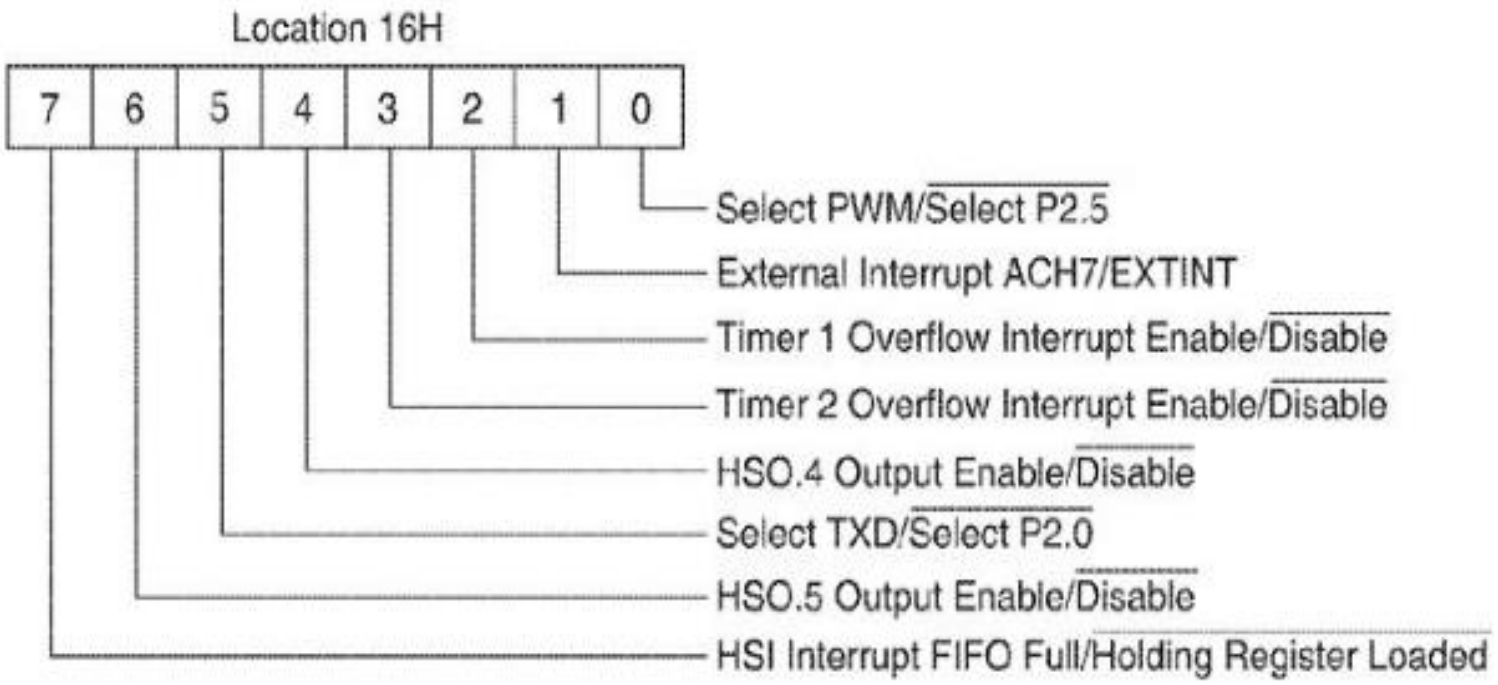
There are two I/O control registers—IOC0 and IOC1

IOC0 is Located at 0015H, this is used to control Timer 2 and High Speed Inputlines.



8096 I/O CONTROL REGISTERS

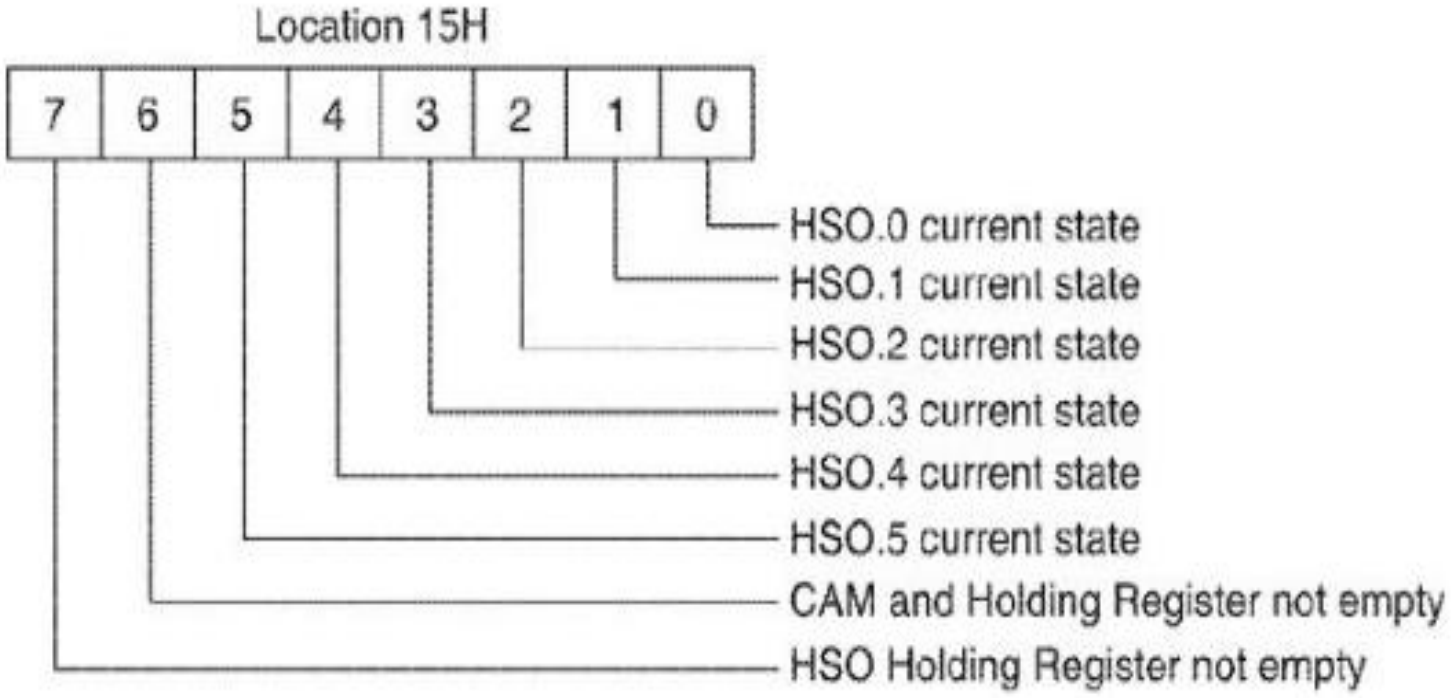
IOC1 is Located at 0016H, this is used to control some interrupt sources, PWMand High Speed Output lines 4 and 5. The format is shown in Figure



8096 I/O STATUS REGISTERS

There are two I/O status registers—IOS0 and IOS1

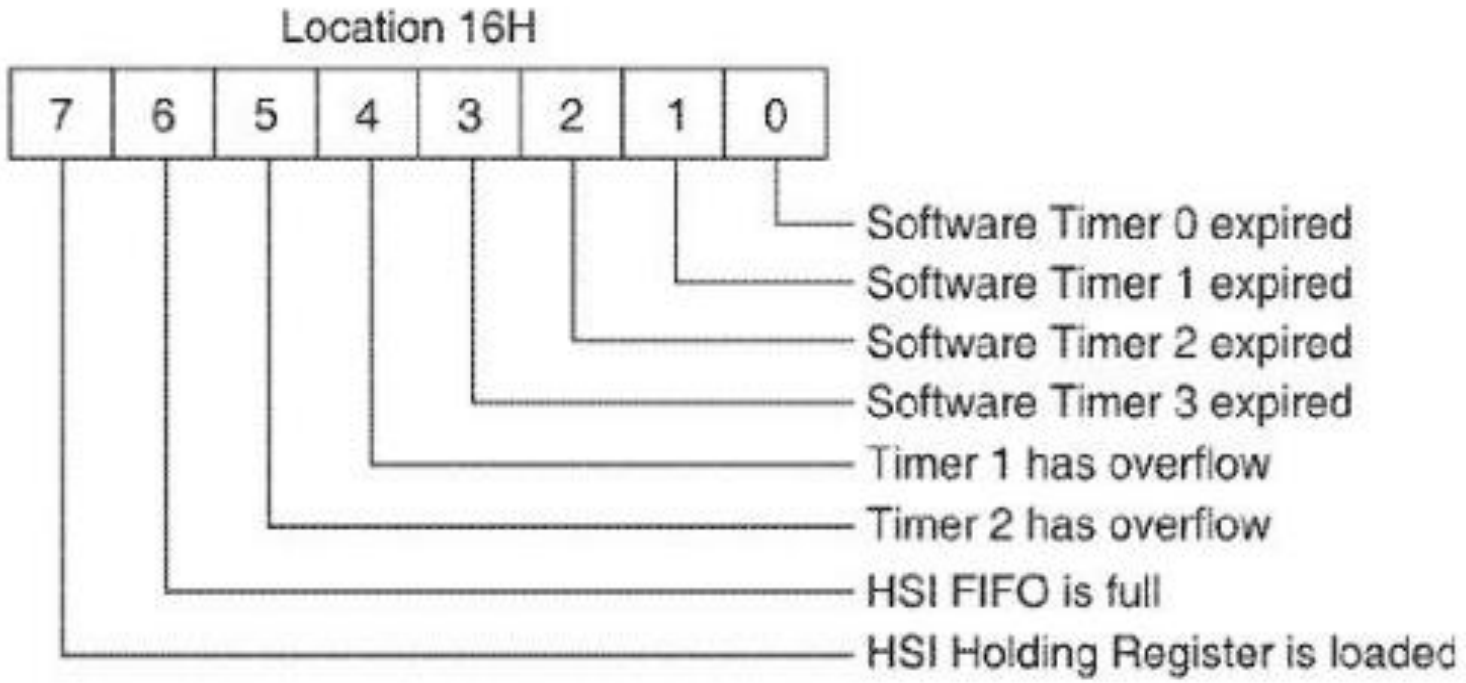
IOS0 Located at 0015H, it holds the current status of the High Speed Output Unit (HSO) lines and Content Addressable Memory CAM



8096 I/O STATUS REGISTERS

There are two I/O status registers—IOS0 and IOS1

IOS1 Located at 0016H, it contains the status bits of Timers and High SpeedInput unit



PROGRAMMER'S MODEL OF THE 8096

The following resources would be used in the programming of the 8096.

Memory

Special Function Registers

I/O Control and Status Registers

Program Status Word

OPERAND TYPES

The 8096 architecture provides support for the following data types.

- Bytes
- Words
- Short Integers
- Integers
- Bits
- Double Words
- Long Integers

- Bytes and short integers are 8-bit variables
- Integer and Words are 16-bit variables.
- However, the 8096 does not allow the direct addressing of bits unlike that in the 8051
- The double words and long integers are 32-bit variables

Operand Addressing

Operands are accessed within the address space of the80C196KB with one of six basic addressing modes.

The six basic address modes which will be described are termed

- Register-direct
- Indirect
- Indirect with auto-increment,
- Immediate,
- short-indexed,
- Long-indexed.

Several other useful addressing operations can be achieved by combining these basic addressing modes with specific registers

Operand Addressing

REGISTER-DIRECT REFERENCES

The register-direct mode is used to directly access a register from the 256 byte on-board register file. The register is selected by an 8-bit field within the instruction and the register address must conform to the operand type's alignment rules. Depending on the instruction, up to three registers can take part in the calculation.

Examples

```
ADD     AX,BX,CX        ; AX:=BX+CX
MUL     AX,BX           ; AX:=AX*BX
INCB    CL              ; CL:=CL+1
```

Operand Addressing

INDIRECT REFERENCES

The indirect mode is used to access an operand by placing its address in a WORD variable in the register file.

The calculated address must conform to the alignment rules for the operand type.

Aninstruction can contain only one indirect reference andthe remaining operands of the instructio

Examples

```
LD    AX,[AX]      ; AX:=MEM_WORD(AX)
ADDB  AL,BL,[CX]   ; AL:=BL+MEM_BYTE(CX)
POP   [AX]         ; MEM_WORD(AX):=MEM_WORD(SP) ; SP:=SP+2
```

Operand Addressing

INDIRECT WITH AUTO-INCREMENT REFERENCES

This addressing mode is the same as the indirect mode except that the WORD variable which contains the indirect address is incremented *after* it is used to address the operand. If the instruction operates on BYTES or SHORT-INTEGERS the indirect address variable will be incremented by one. If the instruction operates on WORDS or INTEGERS the indirect address variable will be incremented by two.

Examples

```
LD    AX,[BX]+     ; AX:=MEM_WORD(BX) ; BX:=BX+2
ADDB  AL,BL,[CX]+  ; AL:=BL+MEM_BYTE(CX) ; CX:=CX+1
PUSH  [AX]+        ; SP:=SP-2;
                   ;   MEM_WORD(SP):=MEM_WORD(AX)
                   ;   AX:=AX+2
```

Operand Addressing

IMMEDIATE REFERENCES

IMMEDIATE REFERENCES

This addressing mode allows an operand to be taken directly from a field in the instruction. For operations on BYTE or SHORT-INTEGER operands this field is eight bits wide. For operations on WORD or INTEGER operands the field is 16 bits wide. An instruction can contain only one immediate reference and the remaining operand(s) must be register-direct references.

Examples

```
ADD  AX,#340 : AX:=AX+340
```

Operand Addressing

SHORT-INDEXED REFERENCES

In this addressing mode an eight bit field in the instruction selects a WORD variable in the register file which contains an address. A second eight bit field in the instruction stream is sign-extended and summed with the WORD variable to form the address of the operand which will take part in the calculation.

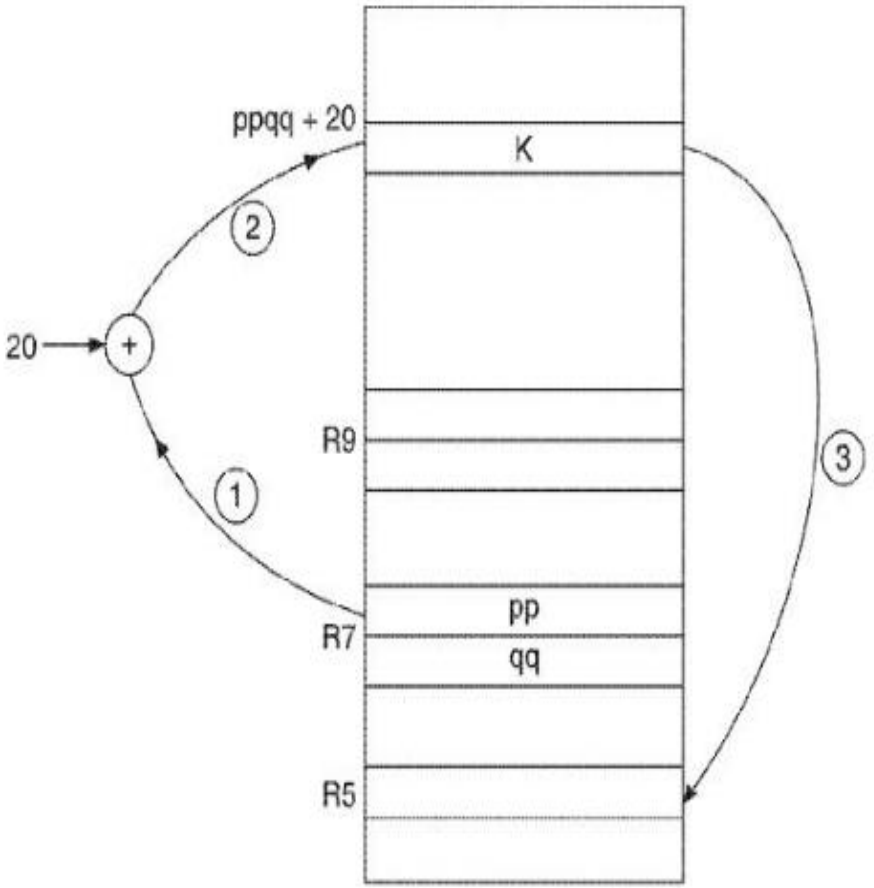
Since the eight bit field is sign-extended, the effective address can be up to 128 bytes before the address in the WORD variable and up to 127 bytes after it. An instruction can contain only one short-indexed reference and the remaining operand(s) must be register-direct references.

LDB R5, 20[R7]

Operand Addressing

SHORT-INDEXED REFERENCES

LDB R5, 20[R7]



Operand Addressing

LONG-INDEXED REFERENCES

This addressing mode is like the short-indexed mode except that a *16-bit* field is taken from the instruction and added to the WORD variable to form the address of the operand. No sign extension is necessary. An instruction can contain only one long-indexed reference and the remaining operand(s) must be register-direct references.

Instruction Sets

ARITHMETIC GROUP

LOGICAL GROUP

SHIFT GROUP

BRANCH GROUP

STACK GROUP

SPECIAL CONTROL GROUP

DATA TRANSFER GROUP

Instruction Sets

ARITHMETIC GROUP: In arithmetic group, the 8096 offers a variety of instructions to performthe operations of addition, subtraction, multiplication, and division.

LOGICAL GROUP:In this group, three logical operations, i.e. AND, OR and XOR aresupported.

SHIFT GROUP: Following types of shift operations are possible inthe 8096.
Left Shift
Logical Right Shift
Arithmetic Right Shift

BRANCH GROUP: In this group, conditional and unconditional jump operations as well assubroutine call and return operations are supported.

STACK GROUP: It is considered that memory address and program status word will bestored in stack as well as retrieved from it. The operations supported arePUSH Word, PUSH Flags, POP Words and POP Flags.

SPECIAL CONTROL GROUP: The operations supported in this group are Clear Bytes, Clear Words, Setand Clear Carry Flags, Enable and Disable Interrupt System, Reset Microprocessor System etc.

Instruction Sets

ARITHMETIC GROUP: In arithmetic group, the 8096 offers a variety of instructions to performthe operations of addition, subtraction, multiplication, and division.

LOGICAL GROUP:In this group, three logical operations, i.e. AND, OR and XOR aresupported.

SHIFT GROUP: Following types of shift operations are possible inthe 8096.
Left Shift
Logical Right Shift
Arithmetic Right Shift

BRANCH GROUP: In this group, conditional and unconditional jump operations as well assubroutine call and return operations are supported.

STACK GROUP: It is considered that memory address and program status word will bestored in stack as well as retrieved from it. The operations supported arePUSH Word, PUSH Flags, POP Words and POP Flags.

SPECIAL CONTROL GROUP: The operations supported in this group are Clear Bytes, Clear Words, Setand Clear Carry Flags, Enable and Disable Interrupt System, Reset Microprocessor System etc.