MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

AY: 2017-18 II SEM CLASS: 2/4 IT (A & B)

SUBJECT: COMPUTER ORGANIZATION AND MICROPROCESSORS (PC 402)

ASSIGNMENT KEY

|  |  |  |  |
| --- | --- | --- | --- |
| S NO | COURSE OUTCOME | QUESTION NUMBER | UNIT |
| 1 | PC 402.1 | 1 | I |
| 2 | PC 402.2 | 2 | II |
| 3 | PC 402.3 | 3 | III |
| 4 | PC 402.4 | 4 | IV |
| 5 | PC 402.5 | 5 | V |

1. a) Distinguish between different Generations of Computer. (Evaluate)

b) Discuss about the Interrupt Hardware. (Understand)

a)

The first generation computers were developed during 1943-1958. It used vacuum tubes as the active electronic components and was therefore very large. However some of the features are as follows-

a) They were extremely large and occupied a very large space.

b) They used vacuum tubes as memory device.

c) They were very expensive and consumed a lot of electrical power.

d) The operating speed was measured in milliseconds.

e) These computers had low level of accuracy and reliability.

f) Storage capacity was too small only 1 to 4Kb.

g) They used machine level programming language.

The examples are- UNIVAC, ENIAC, EDSAC, EDVAC, and UNIVAC.

The second generation computers were developed during 1959-1965. The invention of the transistor by three scientists of Bell Telephone Laboratories in 1947 greatly changed the development of computers. However some of the features are as follows-

a) These computers used transistor.

b) They were smaller, faster and cheaper than first generation of computer.

c) They consumed less electrical power than first generation.

d) The operating speed was measured in microseconds.

e) They were more reliable and accurate than the first generation computers.

f) They could understand high level language such as COBOL.

g) Magnetic tapes were used as secondary storage media.

The examples are – IBM 1620, IBM 1401, and CDC 3600.

The third generation computers were developed during 1966-1973. The development of Integrated Circuit (IC) signaled the beginning of the third generation computers. However some of the features are as follows-

a) These computers used integrated circuits.

b) They were small, efficient and reliable.

c) Operating systems were developed.

d) Monitors and keyboards were introduced for input and output of data.

e) Magnetic disks were used for secondary storage.

f) The operating speed was measured in nano seconds.

g) They could understand large number of high level languages.

The examples are – IBM 360, ICL -1900, and IBM 370 etc.

The fourth generation computers were developed during 1974-1990. This generation of computer is presently in use. The development of microprocessor signaled the beginning of the fourth generation of computers. However some of the features are as follows-

a) These computers use LSI and VLSI technologies.

b) Its sizes were reduced to desktop and laptop computer.

c) These computers are highly reliable and accurate.

d) They have a large memory and high functional speed.

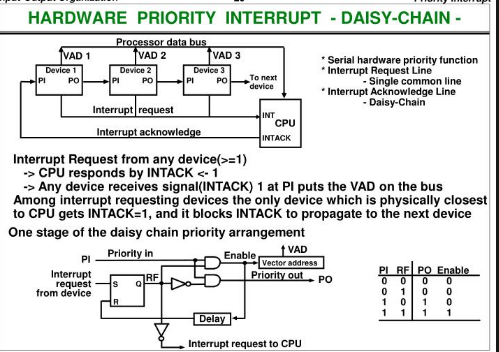
e) The operating speed is measured in beyond picoseconds and MIPS (Million of instruction per second)

f) Magnetic disk is the common source of external storage.

g) Multiprocessing and multiprogramming OS (operating system) are used.

h) 4GL are also used.

b) Discuss about the Interrupt Hardware. (Understand)



1. a) Compare types of RAMs in detail . (Analyze)

Comparison between SRAM (Static Random Access Memory) and DRAM (Dynamic Random Access Memory):

|  |  |  |
| --- | --- | --- |
|  | SRAM | DRAM |
| Definition | It is a type of RAM. SRAM essentially uses latches to store charge. | It is also a type of RAM. DRAM makes use of capacitors to store bits in the form of charge. |
| Speed | Faster | Slower |
| Size | Bigger | Smaller |
| Cost | More expensive per bit | Less expensive per bit |
| Requirement of peripheral circuitary | Comparatively less | Comparatively more |
| Type | Comparatively less common | Comparatively more common |
| Capacity (same technology) | Less | 5 to 10 times more than SRAM |
| Applications | Generally in smaller applications like CPU cache memory and hard drive buffers | Commonly used as the main memory in personal computers |
| Types | Asynchronous SRAM  Synchronous SRAM  Pipeline Burst SRAM | Fast Page Mode DRAM  Extended Data Out DRAM  Burst EDO DRSSM  Synchronous DRAM |
| Access | Easy | Harder |
| Construction | Difficult | Simple |
| Power Consumption | Less | More |
| Density | Low density/less memory per chip | High density/more memory per chip |

Compute the effective memory access time, where cache access time takes 4 ns, while

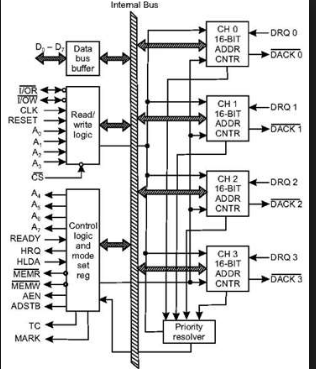
main memory access time is 50 ns with 80% hit ratio.

b) T avg = hc + (1-h)\*m = 13.2 nsec

substitute all the given values

b) Discuss working of 8257 DMA Controller (Understand)

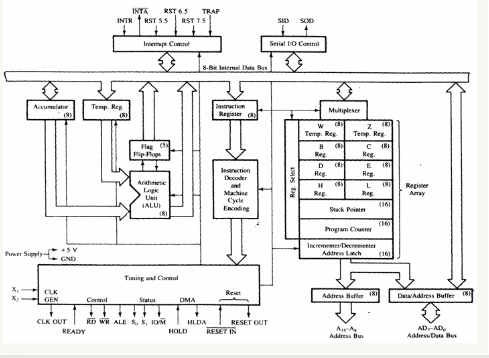
FUNCTIONAL DESCRIPTION General The 8257 is programmable. Direct Memory Access (DMA) device which, when coupled with single Intel® 8212 I/O port device, provides complete four-channel DMA controller for use in Intel® microcomputer systems. After being initialized by software, the 8257 can transfer block of data, containing up to 16.384 bytes, between memory and peripheral device directly, without further intervention required of the CPU. Upon receiving DMA transfer request from an enabled peripheral, the 8257: 1. Acquires control of the system bus. 2. Acknowledges that requesting peripheral which is connected to the highest priority channel. 3. Outputs the least significant eight bits of the memory address onto system address lines A0-A7. outputs the most significant eight bits of the memory address to the 8212 I/O port via.the data bus (the 8212 places these address bits on lines A8-A15), and 4. Generates the appropriate memory and I/O read/ write control signals that cause the peripheral to receive or deposit a data byte directly from or to the addressed location in memory. The 8257 will retain control of the system bus and repeat the transfer sequence, as long as peripheral maintains its DMA request. Thus, the 8257 can transfer block of data to/from high speed peripheral (e.g.. sector of data on floppy disk) in single "burst". When the specified number of data bytes have been transferred, the 8257 activates its Terminal Count (TC) output, informing the CPU that the operation is complete.

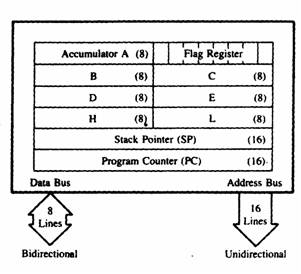


The 8257 offers three different modes of operation: (1) DMA read, which causes data to be transferred from memory to peripheral: (2) DMA write, which causes data to be transferred from peripheral to memory: and (3) DMA verify, which does not actually involve the transfer of data. When an 8257 channel is in the DMA verify mode, it will respond the same as described for transfer operations, except that no memory or I/O read/write control signals will be generated, thus preventing the transfer of data The 8257. however, will gain control of the system bus and will acknowledge the peripheral's DMA request for each DMA cycle. The peripheral can use these acknowledge signals to enable an internal access of each byte of data block in order to execute some verification procedure, such as the accumulation of CRC (Cyclic Redundancy Code) checkword. For example, block of DMA verify cycles might follow block of DMA read cycles (memory to peripheral) to allow the peripheral to verify its newly acquired data. Block Diagram Description 1. DMA Channels The 8257 provides four separate DMA channels (labeled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) DMA address register, and (2) termi nal count register. Both registers must be initialized before channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, terminal count of would cause the TC output to be active in the first DMA cycle for that channel. In general, if the number of desired DMA cycles, load the value N-1 into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel.

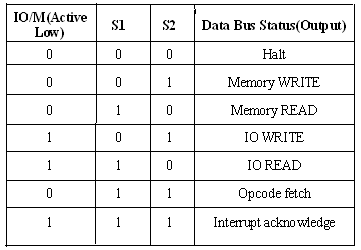
1. a) Explain the Architecture of 8085 with neat figure. ( understand)

This is the functional Block Diagram of 8085 Microprocessor.  
  
Acumulator:-It is a 8-bit register which is used to perform airthmetical and logical operation. It stores the output of any operation. It also works as registers for i/o accesses.



Temporary Register:-It is a 8-bit register which is used to hold the data on which the acumulator is computing operation. It is also called as operand register because it provides operands to ALU.  
  
Registers:-These are general purposes registers. Microprocessor consists 6 general purpose registers of 8-bit each named as B,C,D,E,H and L.   Generally theses registers are not used for storing the data permanently. It carries the 8-bits data. These are used only during the execution of the instructions.  
These registers can also be used to carry the 16 bits data by making the pair of 2 registers. The valid register pairs available are BC,DE HL. We can not use other pairs except BC,DEand HL. These registers are programmed by user.  
  
  
  
ALU:-ALU performs the airthmetic operations and logical operation.  
  
Flag Registers:-It consists of 5 flip flop which changes its status according to the result stored in an accumulator. It is also known as status registers. It is connected to the ALU.  
There are five flip-flops in the flag register are as follows:  
1.Sign(S)  
2.zero(z)  
3.Auxiliary carry(AC)  
4.Parity(P)  
5.Carry(C)  
The bit position of the flip flop in flag register is:  
                 

[http://www.8085projects.info/images/Flags-Pic4.PNG](http://www.8085projects.info/images/Flags-Pic4.PNG)

All of the three flip flop set and reset according to the stored result in the accumulator.  
1.Sign- If D7 of the result is 1 then sign flag is set otherwise reset. As we know that a number on the D7 always desides the sign of the number.  
if D7 is 1: the number is negative.  
if D7 is 0: the number is positive.  
  
2.Zeros(Z)-If the result stored in an accumulator is zero then this flip flop is set otherwise it is reset.  
  
3.Auxiliary carry(AC)-If any carry goes from D3 to D4 in the output then it is set otherwise it is reset.  
  
4.Parity(P)-If the no of 1's is even in the output stored in the accumulator then it is set otherwise it is reset for the odd.  
  
5.Carry(C)-If the result stored in an accumulator generates a carry in its final output then it is set otherwise it is reset.  
  
Instruction registers(IR):-It is a 8-bit register. When an instruction is fetched from memory then it is stored in this register  
Instruction Decoder:- Instruction decoder identifies the instructions. It takes the informations from instruction register and decodes the instruction to be performed.  
  
Program Counter:-It is a 16 bit register used as memory pointer. It stores the memory address of the next instruction to be executed. So we can say that this register is used to sequencing the program. Generally the memory have 16 bit addresses so that it has 16 bit memory.  
The program counter is set to 0000H.  
  
Stack Pointer:-It is also a 16 bit register used as memory pointer. It points to the memory location called stack. Generally stack is a reserved portion of memory where information can be stores or taken back together.  
  
Timing and Control Unit:-It provides timing and control signal to the microprocessor to perform the various operation.It has three control signal. It controls all external and internal circuits. It operates with reference to clock signal.It synchronizes all the data transfers.  
There are three control signal:  
1.ALE-Airthmetic Latch Enable, It provides control signal to synchronize the components of microprocessor.  
2.RD- This is active low used for reading operation.  
3.WR-This is active low used for writing operation.  
  
There are three status signal used in microprocessor S0, S1 and IO/M. It changes its status according the provided input to these pins.  
  
  
                                   
  
Serial Input Output Control-There are two pins in this unit. This unit is used for serial data communication.  
  
Interrupt Unit-There are 6 interrupt pins in this unit. Generally an external hardware is connected to these pins. These pins provide interrupt signal sent by external hardware to microprocessor and microprocessor sends acknowledgement for receiving the interrupt signal. Generally INTA is used for acknowledgement.

**AIM**

To prepare an assembly language program for [**8085**](http://www.circuitstoday.com/tag/8085-lab-manual) to multiply two 8 bit numbers

**PROGRAM**

MVI C,00

LDA 8200

MOV B,A

LDA 8201

MOV A,D

MVI A,00

LABEL:ADD B

DCR D

JNZ LABEL

JNC LOOP

INR C

LOOP:STA 8202

MOV A,C

STA 8203

HLT

**SAMPLE OUTPUT**

|  |  |  |
| --- | --- | --- |
| ADDRESS | DATA | INPUT/OUTPUT |
| 8200 | 04 | Input |
| 8201 | 02 | Input |
| 8202 | 08 | Output |
| 8203 | 00 | Output |

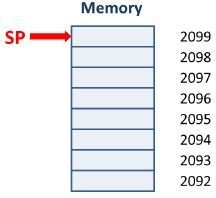
1. a) Explain the concept of Stacks and Subroutines in 8085. (Understand)

A stack is a group of a memory location in the R/W memory that is used for temporary storage of binary information during execution of a program.

The starting memory location of the stack defined in program and space reserved usually at the high end of the memory map.

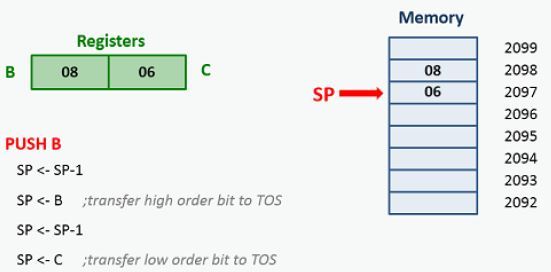
Moreover, The beginning of the stack defined in the program by using instruction **LXI SP, a 16-bit memory address**. Which loads a 16-bit memory address in stack pointer register of the microprocessor.

Once stack location is defined storing of data bytes begins at the memory address that is one less than the address in stack pointer register. LXI SP, 2099h the storing of data bytes, begins at 2098H and continues in reversed numerical order.



Also, Data bytes in register pair of the microprocessor can store on the stack in reverse order by using the PUSH instruction.

PUSH B instruction sore data to register pair BC on a stack.

Stack Subroutines

Moreover, Data bytes can transfer from the stack to respective registers by using instruction POP.

b) Explain the function of Analog to Digital Converter. (Understand)

An ADC converts a continuous-time and continuous-amplitude [analog signal](https://en.wikipedia.org/wiki/Analog_signal) to a [discrete-time](https://en.wikipedia.org/wiki/Discrete-time) and discrete-amplitude [digital signal](https://en.wikipedia.org/wiki/Digital_signal_(signal_processing)). The conversion involves [quantization](https://en.wikipedia.org/wiki/Quantization_(signal_processing)) of the input, so it necessarily introduces a small amount of error or noise. Furthermore, instead of continuously performing the conversion, an ADC does the conversion periodically, [sampling](https://en.wikipedia.org/wiki/Sampling_(signal_processing)) the input, limiting the allowable bandwidth of the input signal.

The performance of ADC is characterized by its [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(signal_processing)) and its [signal-to-noise ratio](https://en.wikipedia.org/wiki/Signal-to-noise_ratio). The bandwidth of an ADC is characterized primarily by its [sampling rate](https://en.wikipedia.org/wiki/Sampling_rate). The [dynamic range](https://en.wikipedia.org/wiki/Dynamic_range)of an ADC is influenced by many factors, including the [resolution](https://en.wikipedia.org/wiki/Audio_bit_depth), linearity and accuracy (how well the quantization levels match the true analog signal), [aliasing](https://en.wikipedia.org/wiki/Aliasing) and [jitter](https://en.wikipedia.org/wiki/Jitter). The dynamic range of an ADC is often summarized in terms of its [effective number of bits](https://en.wikipedia.org/wiki/Effective_number_of_bits) (ENOB), the number of bits of each measure it returns that are on average not [noise](https://en.wikipedia.org/wiki/Noise_(signal_processing)). An ideal ADC has an ENOB equal to its resolution. ADCs are chosen to match the bandwidth and required signal-to-noise ratio

of the signal to be quantized. If an ADC operates at a sampling rate greater than twice the bandwidth of the signal, then [perfect reconstruction](https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem) is possible given an ideal ADC and neglecting quantization error. The presence of quantization error limits the dynamic range of even an ideal ADC. However, if the dynamic range of the ADC exceeds that of the input signal, its effects may be neglected resulting in an essentially perfect digital representation of the input signal.

1. **a) Explain the role of Programmable Peripheral Device (Intel 8255A) (Application)**

The 8255A is a general purpose programmable I/O device designed to transfer the data from I/O to interrupt I/O under certain conditions as required. It can be used with almost any microprocessor.

It consists of three 8-bit bidirectional I/O ports (24I/O lines) which can be configured as per the requirement.

## Ports of 8255A

8255A has three ports, i.e., PORT A, PORT B, and PORT C.

* **Port A** contains one 8-bit output latch/buffer and one 8-bit input buffer.
* **Port B** is similar to PORT A.
* **Port C** can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word.

These three ports are further divided into two groups, i.e. Group A includes PORT A and upper PORT C. Group B includes PORT B and lower PORT C. These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.

## Operating Modes

8255A has three different operating modes −

* **Mode 0** − In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
* **Mode 1** − In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.
* **Mode 2** − In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

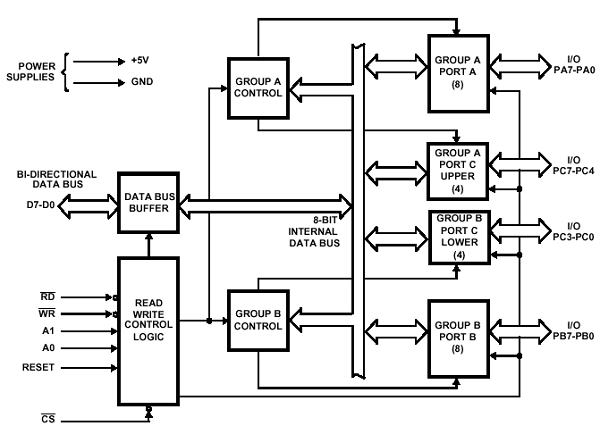
## Features of 8255A

The prominent features of 8255A are as follows −

* It consists of 3 8-bit IO ports i.e. PA, PB, and PC.
* Address/data bus must be externally demux'd.
* It is TTL compatible.
* It has improved DC driving capability.

## 8255 Architecture

The following figure shows the architecture of 8255A −



The Intel 8253 and 8254 are Programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16-bit registers. Each counter has 2 input pins, i.e. Clock & Gate, and 1 pin for “OUT” output. To operate a counter, a 16-bit count is loaded in its register. On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU.

## Difference between 8253 and 8254

The following table differentiates the features of 8253 and 8254 −

|  |  |
| --- | --- |
| **8253** | **8254** |
| Its operating frequency is 0 - 2.6 MHz | Its operating frequency is 0 - 10 MHz |
| It uses N-MOS technology | It uses H-MOS technology |
| Read-Back command is not available | Read-Back command is available |
| Reads and writes of the same counter cannot be interleaved. | Reads and writes of the same counter can be interleaved. |

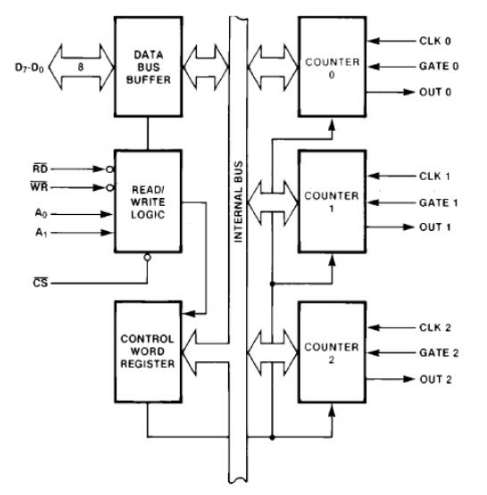
## Features of 8253 / 54

The most prominent features of 8253/54 are as follows −

* It has three independent 16-bit down counters.
* It can handle inputs from DC to 10 MHz.
* These three counters can be programmed for either binary or BCD count.
* It is compatible with almost all microprocessors.
* 8254 has a powerful command called READ BACK command, which allows the user to check the count value, the programmed mode, the current mode, and the current status of the counter.

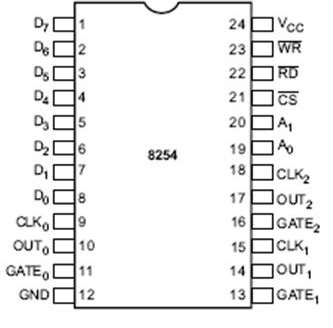
## 8254 Architecture

The architecture of 8254 looks as follows −



## 8254 Pin Description

Here is the pin diagram of 8254 −



In the above figure, there are three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals - CLOCK & GATE, and one output signal - OUT.

### Data Bus Buffer

It is a tri-state, bi-directional, 8-bit buffer, which is used to interface the 8253/54 to the system data bus. It has three basic functions −

* Programming the modes of 8253/54.
* Loading the count registers.
* Reading the count values.

### Read/Write Logic

It includes 5 signals, i.e. RD, WR, CS, and the address lines A0 & A1. In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memorymapped I/O mode, these are connected to MEMR and MEMW.

Address lines A0 & A1 of the CPU are connected to lines A0 and A1 of the 8253/54, and CS is tied to a decoded address. The control word register and counters are selected according to the signals on lines A0 & A1.

|  |  |  |
| --- | --- | --- |
| **A1** | **A0** | **Result** |
| 0 | 0 | Counter 0 |
| 0 | 1 | Counter 1 |
| 1 | 0 | Counter 2 |
| 1 | 1 | Control Word Register |
| X | X | No Selection |

### Control Word Register

This register is accessed when lines A0 & A1 are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A1** | **A0** | **RD** | **WR** | **CS** | **Result** |
| 0 | 0 | 1 | 0 | 0 | Write Counter 0 |
| 0 | 1 | 1 | 0 | 0 | Write Counter 1 |
| 1 | 0 | 1 | 0 | 0 | Write Counter 2 |
| 1 | 1 | 1 | 0 | 0 | Write Control Word |
| 0 | 0 | 0 | 1 | 0 | Read Counter 0 |
| 0 | 1 | 0 | 1 | 0 | Read Counter 1 |
| 1 | 0 | 0 | 1 | 0 | Read Counter 2 |
| 1 | 1 | 0 | 1 | 0 | No operation |
| X | X | 1 | 1 | 0 | No operation |
| X | X | X | X | 1 | No operation |

### Counters

Each counter consists of a single, 16 bit-down counter, which can be operated in either binary or BCD. Its input and output is configured by the selection of modes stored in the control word register. The programmer can read the contents of any of the three counters without disturbing the actual count in process.