MODULE: 5

Interfacing Memory & Parallel I/O Peripherals to DSP Devices

❖ LEARNING OBJECTIVES

- Introduction,
- Memory Space Organization,
- External Bus Interfacing Signals.
- Memory Interface,
- Parallel I/O Interface,
- Programmed I/O,
- Interrupts and I/O
- Direct Memory Access (DMA).

7.1 Introduction:

- A typical DSP system has DSP with external memory, input devices and output devices.
- Since the manufacturers of memory and I/O devices are not same as that of manufacturers of DSP and also since there are variety of memory and I/O devices available,
- The signals generated by DSP may not suit memory and I/O devices to be connected to DSP. Thus, there is a need for interfacing devices the purpose of it being to use DSP signals to generate the appropriate signals for setting up communication with the memory.
- DSP with interface is shown in fig. 7.1.

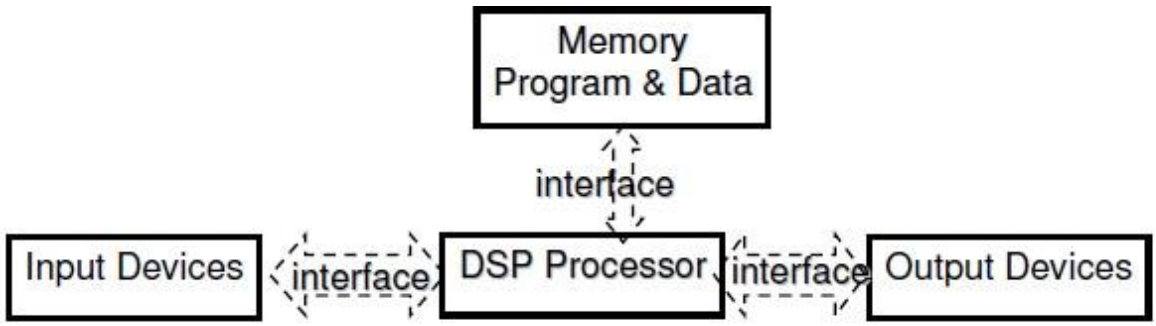


Fig. 7.1: DSP system with interfacing

7.2 Memory Space Organization:

- Memory Space in TMS320C54xx has 192K words of 16 bits each.
- Memory is divided into Program Memory, Data Memory and I/O Space, each are of 64K words.
- The actual memory and type of memory depends on particular DSP device of the family.
- If the memory available on a DSP is not sufficient for an application, it can be interfaced to an external memory as depicted in fig. 7.2.
- The On- Chip Memory are faster than External Memory. There are no interfacing requirements. Because they are on-chip, power consumption is less and size is small.
- It exhibits better performance by DSP because of better data flow within pipeline.
- The purpose of such memory is to hold Program / Code / Instructions, to hold constant data such as filter coefficients / filter order, also to hold trigonometric tables / kernels of transforms employed in an algorithm.
- Not only constants are stored in such memory, they are also used to hold variable data and intermediate results so that the processor need not refer to external memory for the purpose.

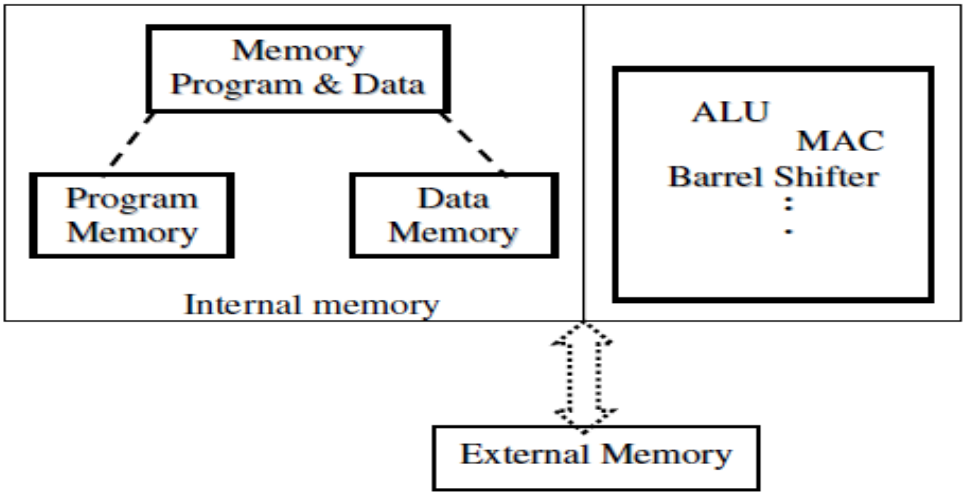


Fig. 7.2: Internal memory and interfacing of external memory

- External memory is off-chip. They are slower memory. External Interfacing is required to establish the communication between the memory and the DSP. They can be with large memory space.
- The purpose is being to store variable data and as scratch pad memory. Program memory can be ROM, Dual Access RAM (DARAM), Single Access RAM (SARAM), or a combination of all these.
- The program memory can be extended externally to 8192K words. That is, 128 pages of 64K words each. The arrangement of memory and DSP in the case of Single Access RAM (SARAM) and Dual Access RAM (DARAM) is shown in fig. 7.3.
- One set of address bus and data bus is available in the case of SARAM and two sets of

address bus and data bus is available in the case of DARAM. The DSP can thus access two memory locations simultaneously.

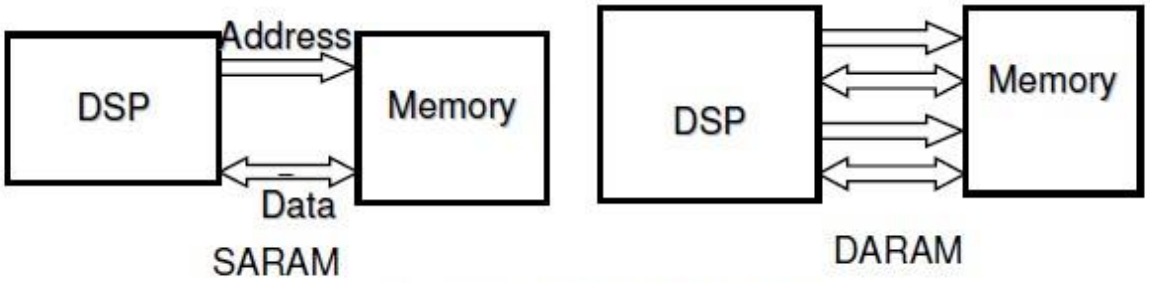


Fig. 7.3: SARAM & DARAM

- There are 3 bits available in memory mapped register, PMST for the purpose of on-chip memory mapping. They are microprocessor / microcomputer mode. If this bit is 0, the on-chip ROM is enabled and addressable and if this bit is 1 the on-chip ROM not available. The bit can be manipulated by software / set to the value on this pin at system reset.
- Second bit is OVLY. It implies RAM Overlay. It enables on-chip DARAM data memory blocks to be mapped into program space. If this bit is 0, on-chip RAM is addressable in data space but not in Program Space and if it is 1, on-chip RAM is mapped into Program & Data Space.
- The third bit is DROM. It enables on-chip DARAM 4-7 to be mapped into data space. If this bit is 0, on-chip DARAM 4-7 is not mapped into data space and if this bit is 1, on-chip DARAM 4-7 is mapped into Data Space. On-chip data memory is partitioned into several regions as shown in table 7.1. Data memory can be onchip / off-chip.

Table 7.1: Data memory 64 K

0000-005F 96 locations	Memory Mapped Registers
0060-007F 32 locations	Scratch pad RAM
0080-7FFF	On-chip DARAM 0-3 32Kx16bit
8000-FFFF 32K locations	On-chip DARAM 4-7 for Data

- The on-chip memory of TMS320C54xx can be both program & data memory.
- It enhances speed of program execution by using parallelism. That is, multiple data access capability is provided for concurrent memory operations.
- The number of operations in single memory access is 3 reads & one write.

- The external memory to DSP can be interfaced with 16 -23 bit Address Bus, 16 bit Data Bus.
- Interfacing Signals are generated by the DSP to refer to external memory. The signals required by the memory are typically chip Select, Output Enable and Write Enable. For example, TMS320C5416 has 16K ROM, 64K DARAM and 64K SARAM.
- Extended external Program Memory is interfaced with 23 address lines i.e., 8192K locations. The external memory thus interfaced is divided into 128 pages, with 64K words per page.

7.3 External Bus Interfacing Signals:

- In DSP there are 16 external bus interfacing signals. The signal is characterized as single bit i.e., single line or multiple bits i.e., Multiple lines / bus.
- It can be synchronous / asynchronous with clock. The signal can be active low / active high. It can be output / input Signal.
- The signal carrying line / lines Can be unidirectional / bidirectional Signal. The characteristics of the signal depend on the purpose it serves. The signals available in TMS320C54xx are listed in table 7.2 (a) & table 7.2 (b).

Table 7.2 (a) External Bus Interfacing Signals

1	A0-A19	20 bit Address Bus
2	D0-D15	16 bit Data Bus
3	$\overline{DS}$	Data Space Select
4	$\overline{PS}$	Program Space Select
5	$\overline{IS}$	I/O Space Select
6	R / $\overline{W}$	Read/Write Signal
7	$\overline{MSTRB}$	Memory Strobe
8	$\overline{IOTRB}$	I/O Strobe

- In external bus interfacing signals, address bus and data bus are multi-lines bus. Address bus is unidirectional and carries address of the location refereed.
- Data bus is bidirectional and carries data to or from DSP. When data lines are not in use, they are tri-stated. Data Space Select, Program Space Select, I/O Space Select are meant for data space, program space or I/O space selection.
- These interfacing signals are all active low. They are active during the entire operation of data memory / program memory / I/O space reference.
- Read/Write Signal determines if the DSP is reading the external device or writing.Read/Write Signal is low when DSP is writing and high when DSP is reading

- Strobe Interfacing Signals, Memory Strobe and I/O Strobe both are active low. They remain low during the entire read & write operations of memory and I/O operations respectively.
- External Bus Interfacing Signals from 1-8 are all are unidirectional except Data Bus which is bidirectional. Address Lines are outgoing signals and all other control signals are also outgoing signals.

Table 7.2 (b) External Bus Interfacing Signals

9	READY	Data Ready Signal
10	$\overline{\text{HOLD}}$	Hold Request
11	$\overline{\text{HLDA}}$	Hold Acknowledge
12	$\overline{\text{MSC}}$	Micro State Complete
13	$\overline{\text{IRQ}}$	Interrupt Request
14	$\overline{\text{IACK}}$	Interrupt Acknowledge
15	XF	External Flag Output
16	$\overline{\text{BIO}}$	Branch Control Input

- Selects one or more memory ICs among many memory ICs in the system. Write Enable enables writing of data available on data bus to a memory location.
- Output Enable signal enables the availability Data Ready signal is used when a slow device is to be interfaced.
- Hold Request and Hold Acknowledge are used in conjunction with DMA controller.
- There are two Interrupt related signals: Interrupt Request and Interrupt Acknowledge. Both are active low.
- Interrupt Request typically for data exchange. For example, between ADC / another Processor.
- TMS320C5416 has 14 hardware interrupts for the purpose of User interrupt, Mc-BSP, DMA and timer.
- The External Flag is active high, asynchronous and outgoing control signal. It initiates an action or informs about the completion of a transaction to the peripheral device. Branch Control Input is a active low, asynchronous, incoming control signal. A low on this signal makes the DSP to respond or attend to the peripheral device. It informs about the completion of a transaction to the DSP.

7.4 The Memory Interface:

- The memory is organized as several locations of certain number of bits.
- The number of locations decides the address bus width and memory capacity.
- The number of bits per locations decides the data bus width and hence word length. Each location has unique address.
- The demand of an application may be such that memory capacity required is more than that available in a memory IC. That means there are insufficient words in memory IC. Or the word length required may be more than that is available in a memory IC. Thus, there may be insufficient word length. In both the cases, more number of memory ICs are required.
- Typical signals in a memory device are address bus to carry address of referred memory location. Data bus carries data to or from referred memory location.
- Chip Select Signal of data from a memory location onto the data bus. The address bus is unidirectional, carries address into the memory IC.
- Data bus is bidirectional. Chip Select, Write Enable and Output Enable control signals are active high or low and they carry signals into the memory ICs.
- The task of the memory interface is to use DSP signals and generate the appropriate signals for setting up communication with the memory.
- The logical spacing of interface is shown in fig. 7.4.

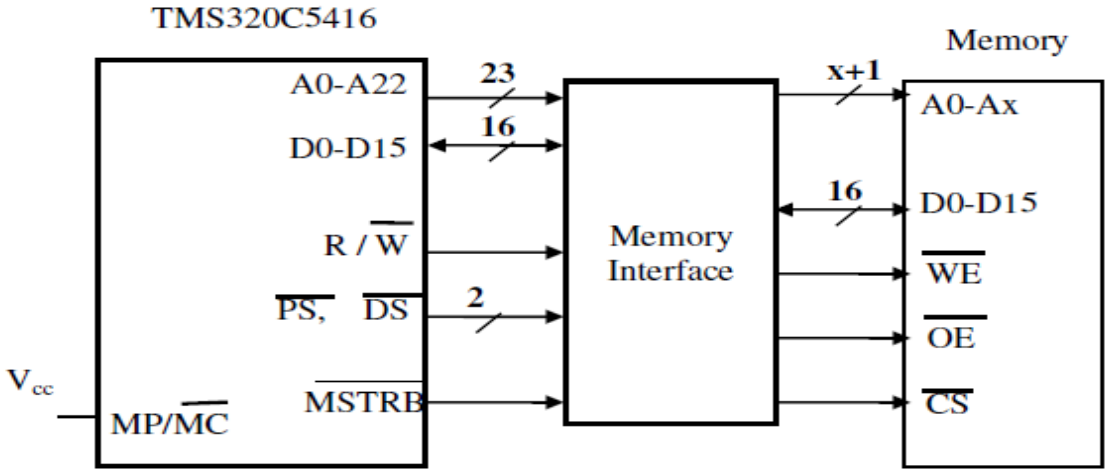


Fig. 7.4 Memory Interface for TMS320C5416

- The timing sequence of memory access is shown in fig. 7.5. There are two read operations, both referring to program memory. Read Signal is high and Program Memory Select is low.
- There is one Write operation referring to external data memory.
- Data Memory Select is low and Write Signal low. Read and write are to memory device and hence memory strobe is low.
- Internal program memory reads take one clock cycle and External data memory access require two clock cycles.

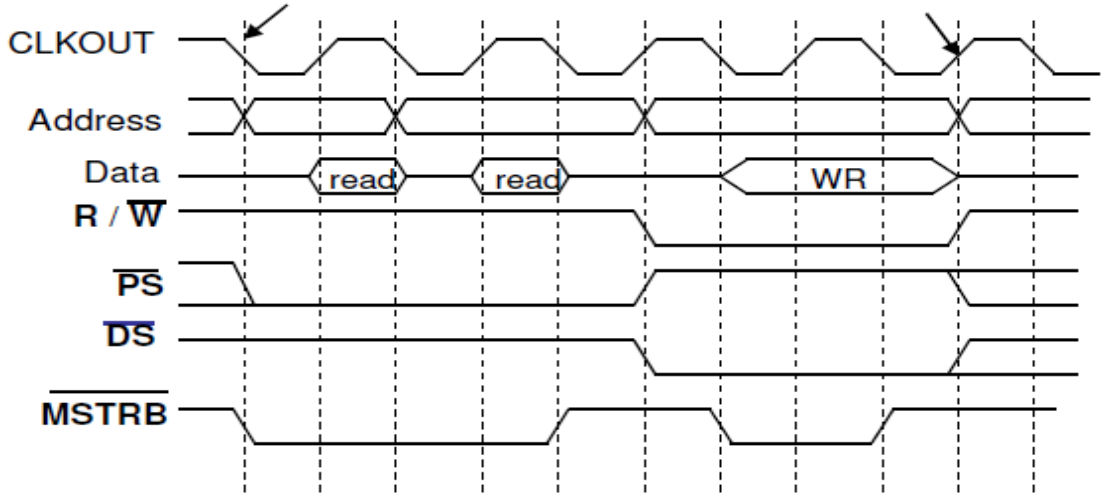


Fig. 7.5 Timing Sequence for External Memory Access

- Effects of 'No decode' interface are
  - Fast memory Access
  - ENTIRE Address space is used by the Device that is connected
  - Memory responds to 0000-1FFFh and also to all combinations of address bits A13- A19 (In the example quoted)
  - Program space select & data space select lines are not used
  - SRAM is thus indistinguishable as a program or data Memory

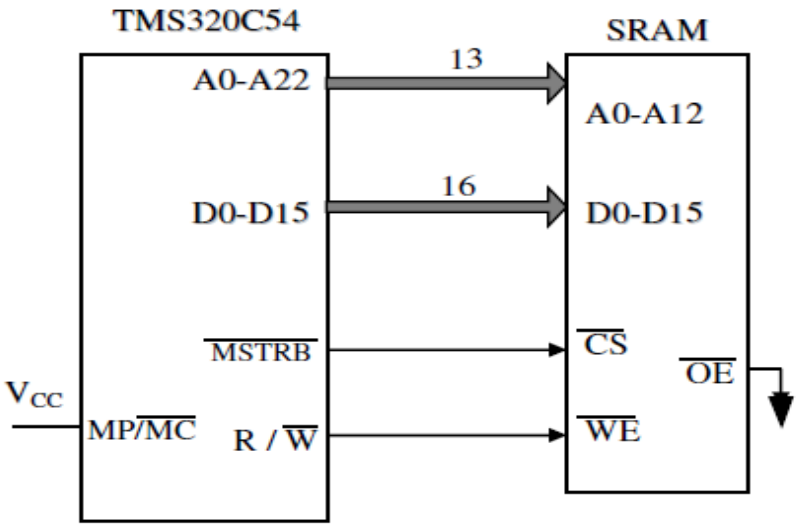


Fig. P7.4: Memory interface without decode circuit

PROBLEM NO 7.1: Design an interface to connect a 64K × 16 flash memory to a TMS320C54XX device. the processor address bus to be used is A0 – A15. The flash memory has the signal as shown in fig.p7.5

SOLUTION:

Address lines from A0-A15 are used to addresses 64 location. All the data lines. D0-D15 are used to carry data word. A data space select line is connected to chip enable of memory so that whenever DSP refer to data memory. This flash memory is enabled. When DSP refers to memory and it is a write operation. Both memory strobe and read/write signal will be low. They are combined in using OR gate and used as write enable for memory. Memory read is performed by combining memory strobe and XF signals.

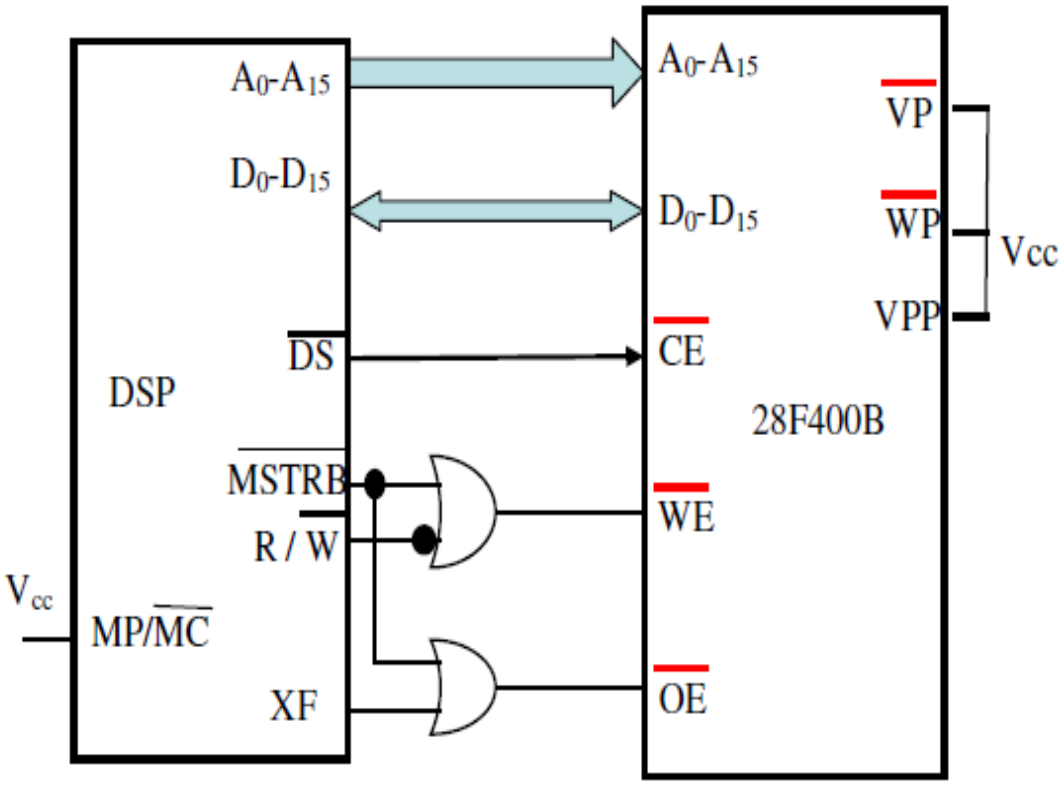


Fig. P7.5: Interfacing flash memory

PROBLEM NO:7.2

What is the range of addresses that can be decoded if A19 is pulled low in a processor with 20 address lines?

SOLUTION: Address line : A0 – A19

The 20 address lines are (A0- A190 with A19 is pulled low, effectively there are 19 address lines. These numbers of addressable location is

$$2^{20} = 80000h$$

The range of address is 00000-7FFFFh

PROBLEM NO 7.3:

With a neat schematic diagram, Design a data memory system with addresses rang 000800h – 000FFFh for a C5416 processor. Use 2k × 8 SRAM memory chips.

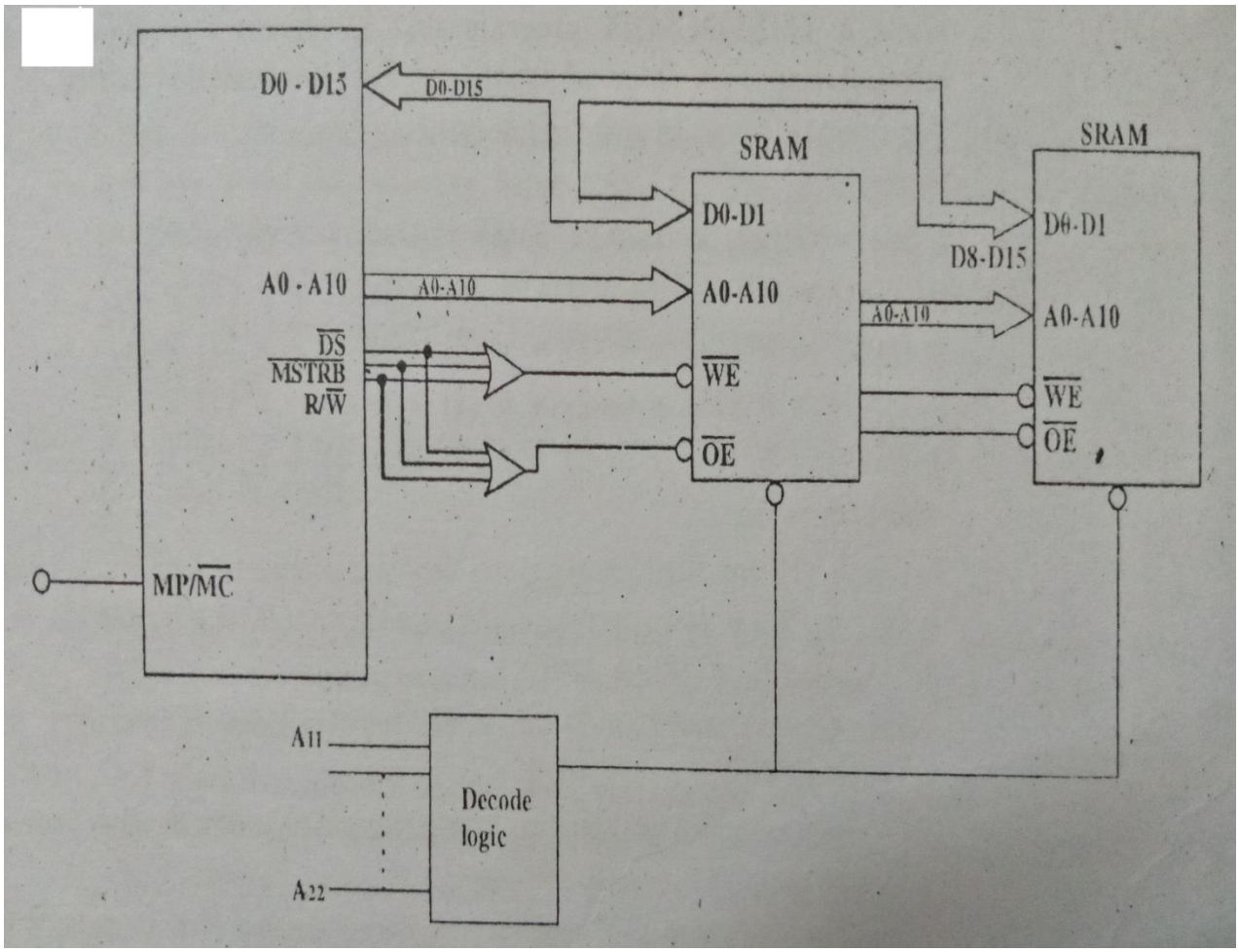
SOLATION:

i) **The width of the data bus for memory chips is 8 bits, but the width of the data bus for the processors is 16 bits.**

ii) **D0-D7 of the processor is connected to D0-D1 of the first memory chip and D8-D15 to D0-D7 of the second memory chip to create the 16 bit data bus.**

iii) **Output enable and write enable for the memory chips are generated by combining the $\overline{DS}$, $\overline{MSTRB}$ and R/$\overline{W}$**

iv) **Address lines A1 – A22 are used in the decode logic to generate chip select signals for the memory devices. These must all be logic 0 to generate the chip select for the two device so that the memory respond to the desired address range.**



7. a. **Design a circuit to interface 4K × 16 and 2K × 16 memory chips to realize program memory space for TMS320C54XX DSP in the address range 03F000H – 03FFFH and 05F800H – 05FFFFH respectively. Plot the memory mapping.** **(12 Marks)**

Ans. **Memory chip :** $4KX16 = 2^{12} \times 16$

03F000 to 03FFFH

Address Bus : 23 bit (A_{22} to A_0)

$A_{22}\ A_{21}\ A_{20}\ \ A_{19}\ A_{18}\ A_{17}\ A_{16}\ \ A_{15}\ A_{14}\ A_{13}\ A_{12}\ \ A_{11}\ A_{10}\ A_9\ A_8\ \ A_7\ A_6\ A_5\ A_4\ \ A_3\ A_2\ A_1\ A_0$

03F000H = 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

03FFFFH = 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

12 address lines are used – (A_0 to A_{11})

11 address lines are left – ($A_{12} - A_{22} = 1$) and ($A_{18} - A_{22} = 0$)

Memory chip : $2K \times 16 = 2^{11} \times 16$

05F800 – 05FFFFH

$A_{22}\ A_{21}\ A_{20}\ \ A_{19}\ A_{18}\ A_{17}\ A_{16}\ \ A_{15}\ A_{14}\ A_{13}\ A_{12}\ \ A_{11}\ A_{10}\ A_9\ A_8\ \ A_7\ A_6\ A_5\ A_4\ \ A_3\ A_2\ A_1\ A_0$

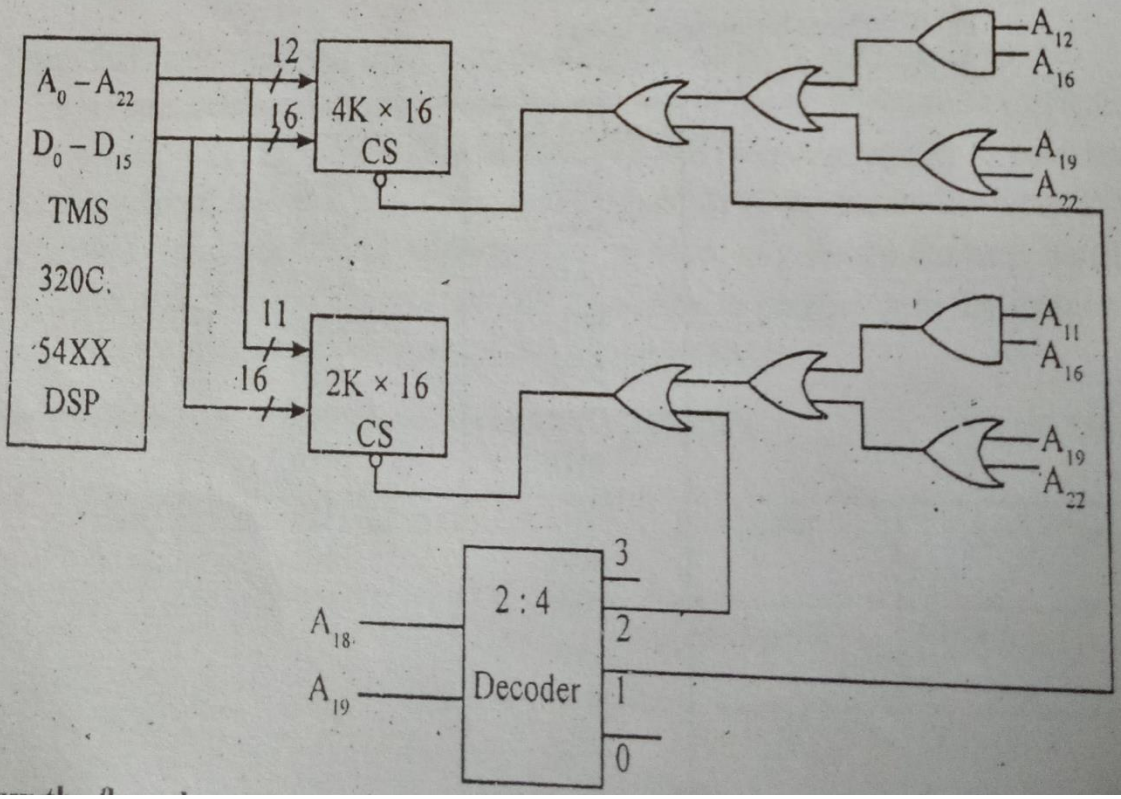
05F800H = 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

05FFFFH = 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

11 address lines are needed → ($A_0 - A_{10}$)

12 address lines are left → ($A_{11} - A_{22}$) out of which $A_{17} = A_{19} = A_{22} = 0$ and A_{11} to $A_{16} = A_{18} = 1$

Block diagram:-



7.5 Parallel I/O Interface:

- I/O devices are interfaced to DSP using unconditional I/O mode, programmed I/O mode or interrupt I/O mode. Unconditional I/O does not require any handshaking signals.
- DSP assumes the readiness of the I/O and transfers the data with its own speed. Programmed I/O requires handshaking signals.
- DSP waits for the readiness of the I/O readiness signal which is one of the handshaking signals. After the completion of transaction DSP conveys the same to the I/O through another handshaking signal.
- Interrupt I/O also requires handshaking signals. DSP is interrupted by the I/O indicating the readiness of the I/O.
- DSP acknowledges the interrupt, attends to the interrupt. Thus, DSP need not wait for the I/O to respond. It can engage itself in execution as long as there is no interrupt.

7.6 Programmed I /O interface:

- The timing diagram in the case of programmed I/O is shown in fig.7.6. I/O strobe and I/O space select are issued by the DSP. Two clock cycles each are required for I/O read and I/O write operations.

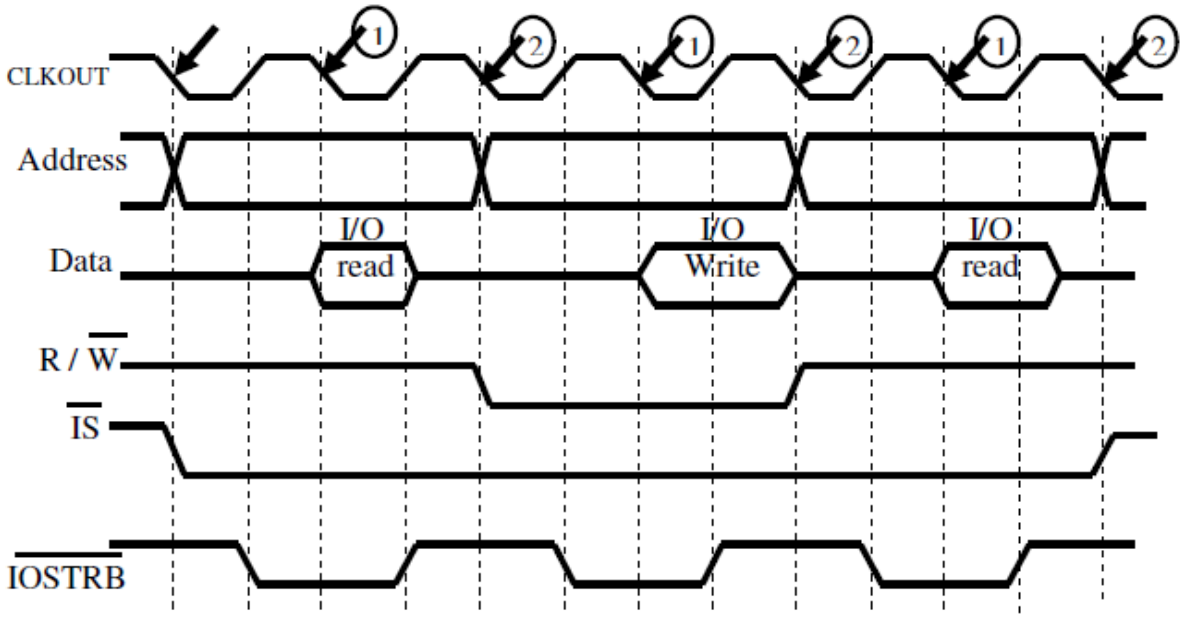


Fig. 7.6: Read-Write-Read Sequence of Operations

- An example of interfacing ADC to DSP in programmed I/O mode is shown in fig. 7.7.
- ADC has a start of conversion (SOC) signal which initiates the conversion. In programmed I/O mode, external flag signal is issued by DSP to start the conversion. ADC issues end of conversion (EOC) after completion of conversion.
- DSP receives Branch input control by ADC when ADC completes the conversion. The

DSP issues address of the ADC, I/O strobe and read / write signal as high to read the data.

- An address decoder does the translation of this information into active low read signal to ADC.
- The data is supplied on data bus by ADC and DSP reads the same. After reading, DSP issues start of conversion once again after the elapse of sample interval.
- Note that there are no address lines for ADC. The decoded address selects the ADC. During conversion, DSP waits checking branch input control signal status for zero.
- The flow chart of the activities in programmed I/O is shown in fig. 7.8.

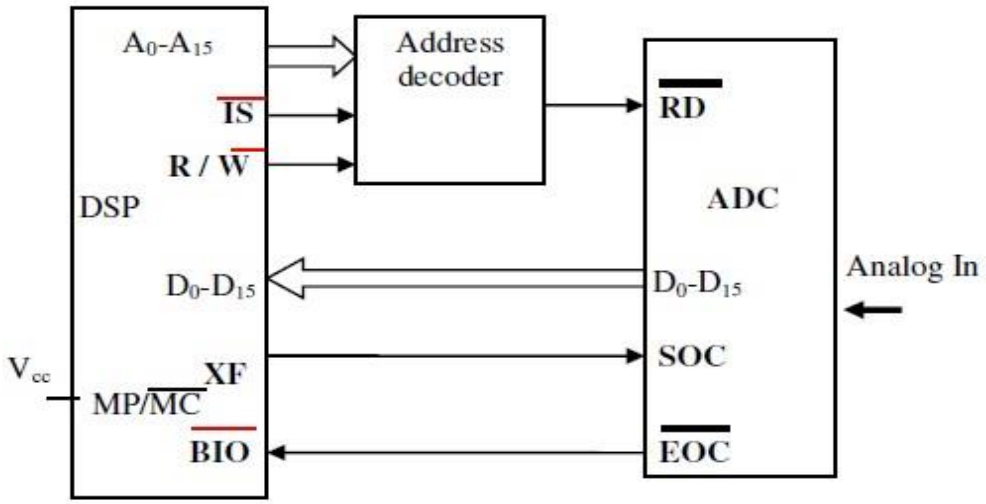


Fig. 7.7: ADC in Programmed I/O mode

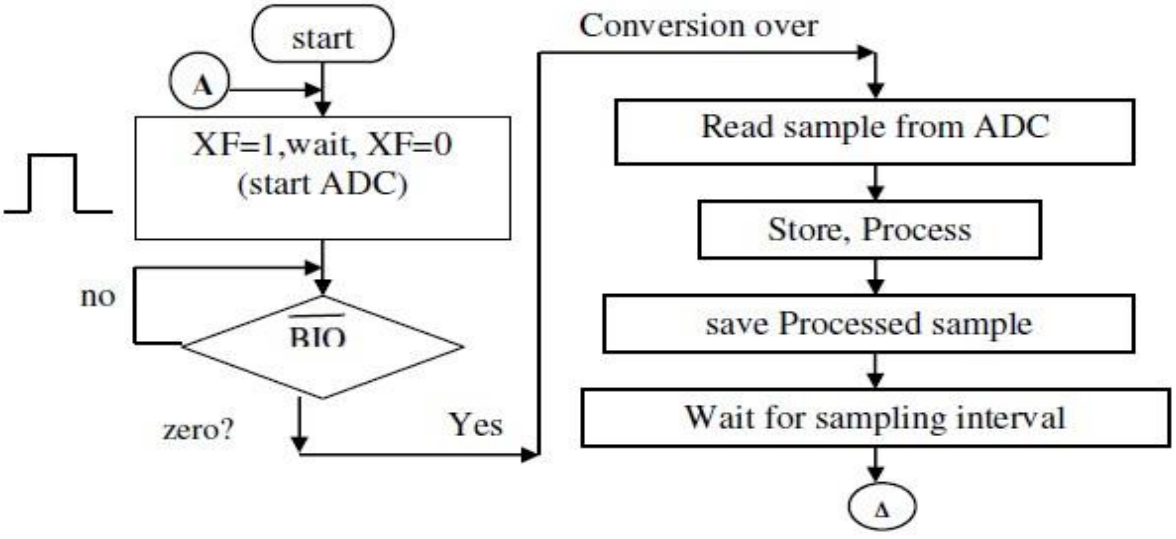


Fig. 7.8: Programmed I/O mode

7.7 Interrupt and I/O:

- This mode of interfacing I/O devices also requires handshaking signals. DSP is interrupted by the I/O whenever it is ready.
- DSP Acknowledges the interrupt, after testing certain conditions, attends to the interrupt. DSP need not wait for the I/O to respond. It can engage itself in execution.
- There are a variety of interrupts. One of the classifications is maskable and nonmaskable. If makeable, DSP can ignore when that interrupt is masked.
- Another classification is vectored and non- vectored. If vectored, Interrupt Service subroutine (ISR) is in specific location. In Software Interrupt, instruction is written in the program.
- In Hardware interrupt, a hardware pin, on the DSP IC will receive an interrupt by the external device. Hardware interrupt is also referred to as external interrupt and software interrupt is referred to as internal interrupt.
- Internal interrupt may also be due to execution of certain instruction can causing interrupt. In TMS320C54xx there are total of 30 interrupts. Reset, Non-mask able, Timer Interrupt, HPI, one each, 14 Software Interrupts, 4 External user Interrupts, 6 Mc-BSP related Interrupts and 2 DMA related interrupts.
- Host Port Interface (HPI) is an 8 bit parallel port. It is possible to interface to a Host Processor using HPI. Information exchange is through on-chip memory of DSP which is also accessible Host processor.

7.7.1 Handling of Interrupts:

- Registers used in managing interrupts are Interrupt flag Register (IFR) and Interrupt Mask Register (IMR).
- IFR maintains pending external & internal interrupts. One in any bit position implies pending interrupt. Once an interrupt is received, the corresponding bit is set.
- IMR is used to mask or unmask an interrupt. One implies that the corresponding interrupt is unmasked. Both these registers are Memory Mapped Registers. One flag, Global enable bit (INTM), in ST1 register is used to enable or disable all interrupts globally. If INTM is zero, all unmasked interrupts are enabled. If it is one, all maskable interrupts are disabled.
- When an interrupt is received by the DSP, it checks if the interrupt is maskable. If the interrupt is non-mask able, DSP issues the interrupt acknowledgement and thus serves the interrupt.
- If the interrupt is hardware interrupt, global enable bit is set so that no other interrupts are entertained by the DSP.
- If the interrupt is maskable, status of the INTM is checked. If INTM is 1, DSP does not respond to the interrupt and it continues with program execution. If the INTM is 0, bit in

IMR register corresponding to the interrupt is checked. If that bit is 0, implying that the interrupt is masked,

- DSP does not respond to the interrupt and continues with its program execution. If the interrupt is unmasked, then DSP issues interrupt acknowledgement. Before branching to the interrupt service routine,
- DSP saves the PC onto the stack. The same will be reloaded after attending the interrupt so as to return to the program that has been interrupted. The response of DSP to an Interrupt is shown in flow chart in fig. 7.9.

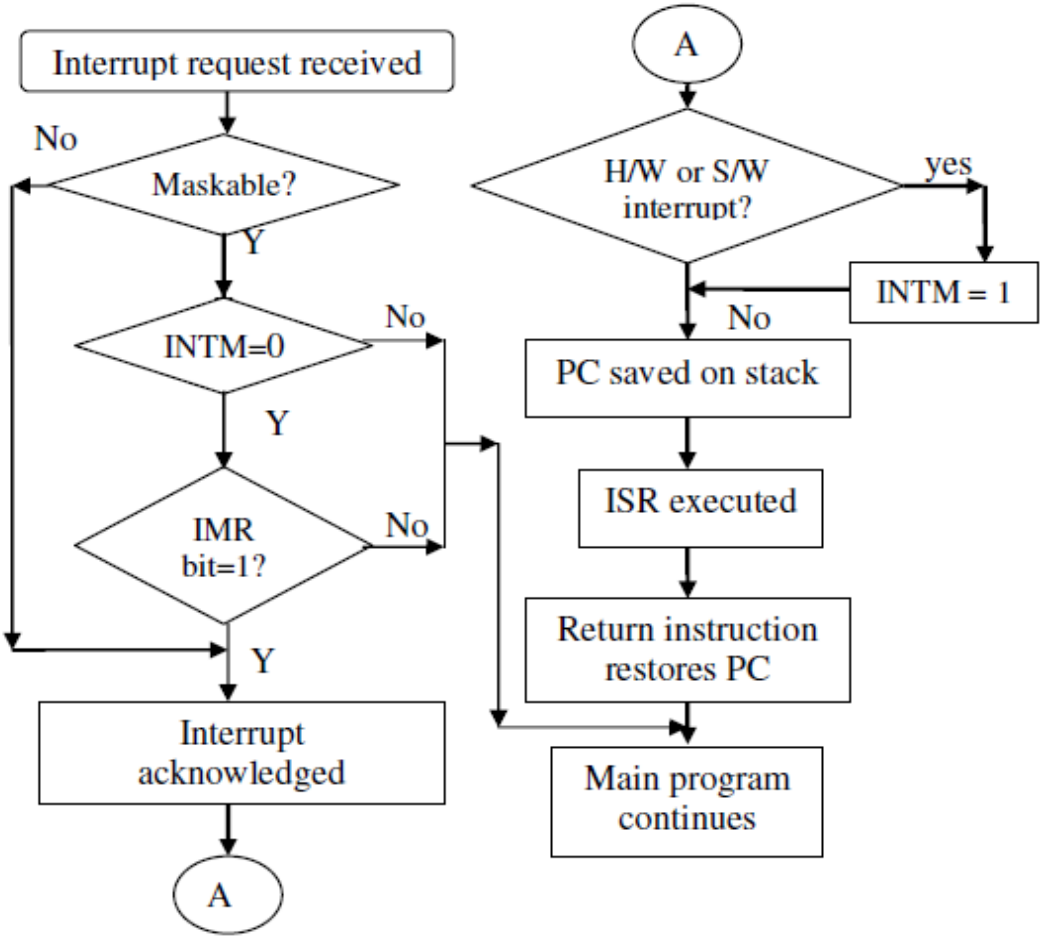
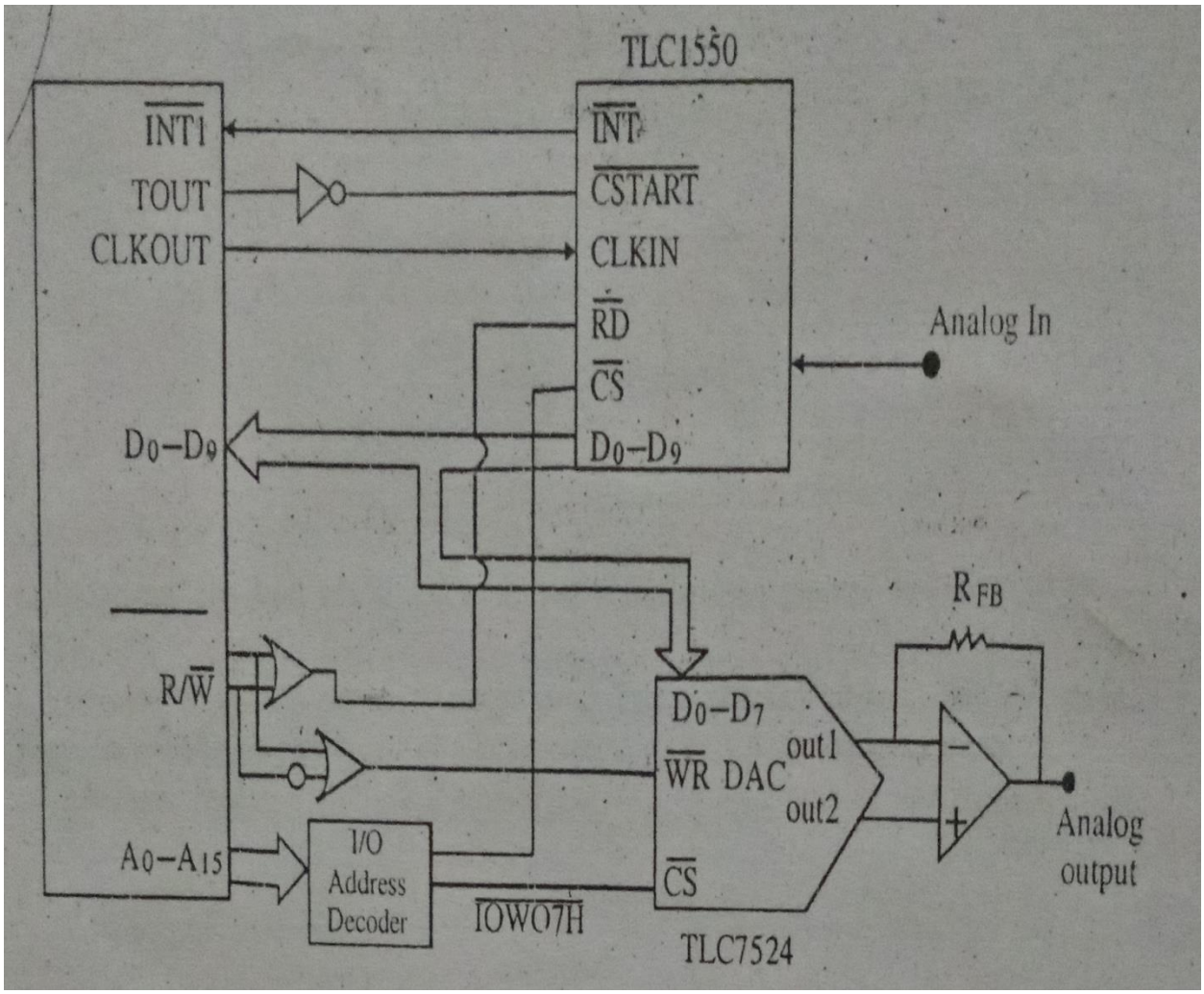


Fig. 7.9: Response of DSP to interrupt

PROBLEM NO: 7.4

Interface the TMS320C54XX to 10 bit ADC (TCL 1550) and an 8 bit DAC (TCL 7524). The sampled signal read from the ADC is to be written to the DAC after adjusting its size. The start of the conversation is to be initiated by the TOUT signal of the timer.

Solution:



The ADC and DAC can be connected to the DSP as shown in the figure. The rate of generation of TOUT is the sampling for the ADC. Conversion is initiated by TOUT and soon as it is completed, INT goes low and the DSP services the interrupt by initiating the execution of the ISR for INT1. The interrupt services routine involves the reading of the sampled data from the port for the ADC data and writing it to the port for the DAC. Before writing the data to the output port, it is shifted to the right by 2 bits, because the output from the ADC is a 10 bit word where the DAC can receive only 8 bits word.

7.8 Direct Memory Access (DMA) operation:

- In any application, there is data transfer between DSP and memory and also DSP and I/O device, as shown in fig. 7.10. However, there may be need for transfer of large amount of data between two memory regions or between memory and I/O.
- DSP can be involved in such transfer, as shown in fig. 7.11. Since amount of data is large, it will engage DSP in data transfer task for a long time.
- DSP thus will not get utilized for the purpose it is meant for, i.e., data manipulation. The intervention of DSP has to be avoided for two reasons: to utilize DSP for useful signal processing task and to increase the speed of transfer by direct data transfer between memory or memory and I/O.
- The direct data transfer is referred to as direct memory access (DMA). The arrangement expected is shown in fig. 7.12. DMA controller helps in data transfer instead of DSP.
-

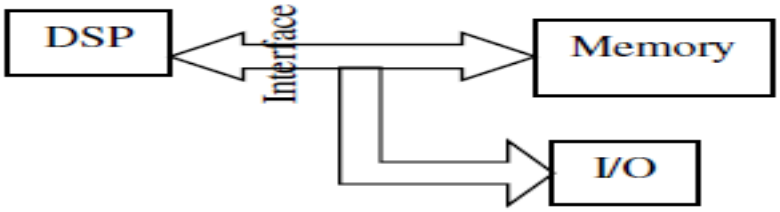


Fig. 7.10: Interface between DSP and external devices

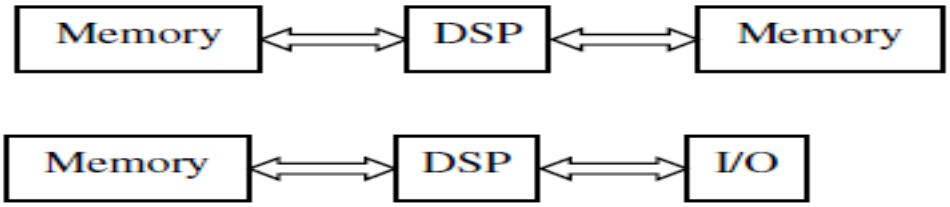


Fig. 7.11: Data transfer with intervention by DSP

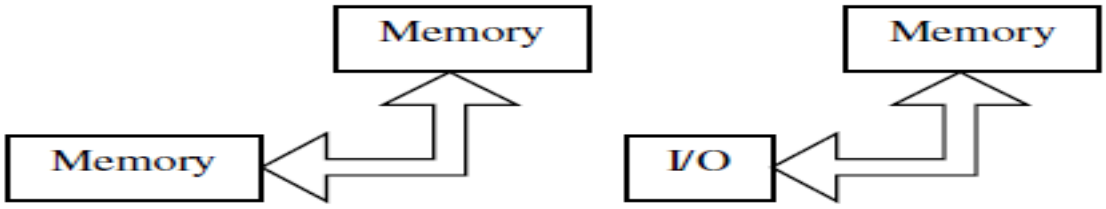


Fig. 7.12: data transfer without intervention by DSP

- In DMA, data transfer can be between memory and peripherals which are either internal or external devices.
- DMA controller manages DMA operation. Thus DSP is relieved of the task of data transfer. Because of direct transfer, speed of transfer is high.
- In TMS320C54xx, there are up to 6 independent programmable DMA channels. Each channel is between certain source & destination.
- One channel at a time can be used for data transfer and not all six simultaneously.
- These channels can be prioritized. The speed of transfer measured in terms of number of clock cycles for one DMA transfer depends on several factors such as source and destination location, external interface conditions, number of active DMA channels, wait states and bank switching time.
- The time for data transfer between two internal memories is 4 cycles for each word.
- Requirements of maintaining a channel are source & Destination address for a channel, separately for each channel.
- Data transfer is in the form of block, with each block having frames of 16/ 32 bits. Block size, frame size, data are programmable. Along with these, mode of transfer and assignment of priorities to different channels are also to be maintained for the purpose of data transfer.
- There are five, channel context registers for each DMA channel.
- They are
    - Source Address Register (DMSRC),
    - Destination Address Register (DMDST),
    - Element Count Register (DMCTR),
    - Sync select & Frame Count register (DMSFC),
    - Transfer Mode Control Register (DMMCR).
- There are four reload registers. The context register DMSRC & DMDST are source & destination address holders.
- DMCTR is for holding number of data elements in a frame. DMSFC is to convey sync event to use to trigger DMA transfer, word size for transfer and for holding frame count.
- DMMCR Controls transfer mode by specifying source and destination spaces as program memory, data memory or I/O space.
- Source address reload & Destination address reload are useful in reloading source address and destination address. Similarly, count reload and frame count reload are used in reloading count and frame count.
- Additional registers for DMA that are common to all channels are Source Program page address, DMSRCP, Destination Program page address, DMDSTP, Element index address register, Frame index address register.
- Number of memory mapped registers for DMA is 6x (5+4) and some common registers for all channels, amounting to total of 62 registers required. However, only 3 (+1 for priority related) are available.
- They are DMA Priority & Enable Control Register (DMPREC), DMA sub bank Address Register (DMSA), DMA sub bank Data Register with auto increment (DMSDI) and DMA

sub bank Data Register (DMSDN).

- To access each of the DMA Registers Register sub addressing Technique is employed. The schematic of the arrangement is shown in fig. 7.13.
- A set of DMA registers of all channels (62) are made available in set of memory locations called sub bank. This voids the need for 62 memory mapped registers.
- Contents of either DMSDI or DMSDN indicate the code (1's & 0's) to be written for a DMA register and contents of DMSA refers to the unique sub address of DMA register to be accessed.
- Mux routes either DMSDI or DMSDN to the sub bank. The memory location to be written DMSDI is used when an automatic increment of the sub address is required after each access.
- Thus it can be used to configure the entire set of registers. DMSDN is used when single DMA register access is required.
- The following examples bring out clearly the method of accessing the DMA registers and transfer of data in DMA mode.

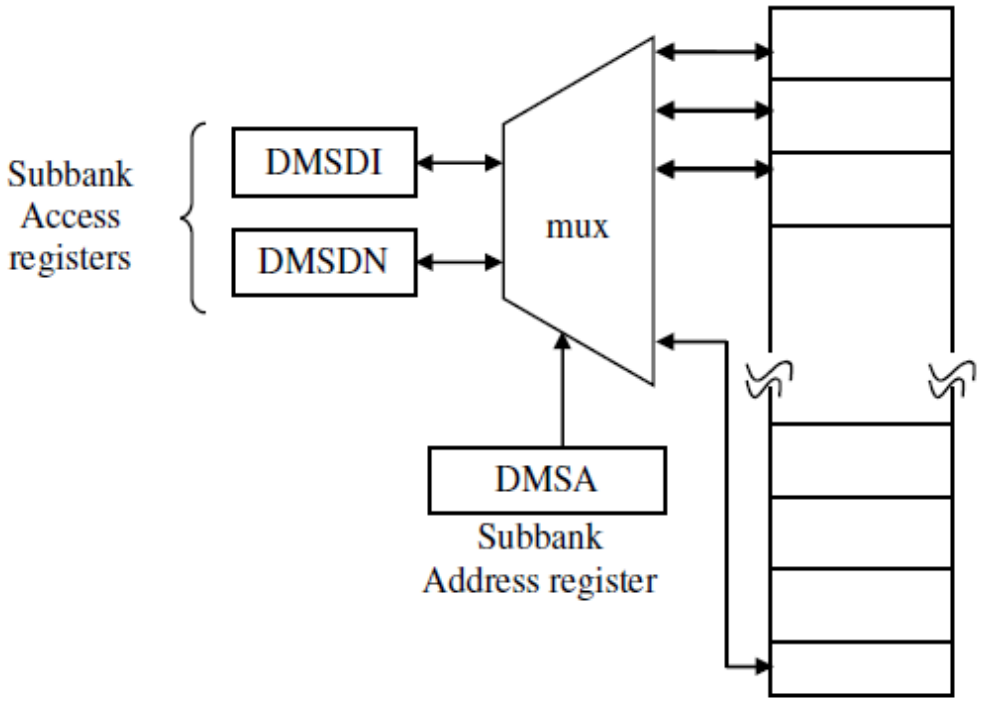


Fig. 7.13: Register Subaddress Technique

ASSIGNMENT NO: 7 (Recommended Questions)

1. Explain an interface between an A/D converter and the TMS320C54XX processor in the programmed I/O mode.
2. Explain how the interrupts are handled in TMS32054XX processor, with the help of a flow chart.[**DEC-2011,8M**]
3. Explain briefly memory space organization in TMS32054XX memory.[**DEC-2011,4M**]
4. Sketch the i/o interface at the pins R/W,IS and IOSTRB for a read-write-read sequence of operations.[**june/july 2011,8M**]
5. With a neat schematic diagram design a data memory system with address range 000800h-000FFFh for a C5416 processor. Use 2k×8 SRAM memory chips.[**DEC-2011,8M]**
6. Describe DMA with respect to TMS320C54XX processors. .[**DEC-2011,4M]**
7. Drew the timing diagram for memory interface for read-read-write sequence of operation. Explain the purpose of each signal involved.
8. Explain the memory interface block diagram for the TMS 320 C54xx processor**.**
9. Draw the I/O interface timing diagram for read – write read sequence of operation.
10. What are interrupts? How interrupts are handled by C54xx DSP Processors.
11. Explain the memory interface block diagram for the TMS 320 C54xx processor.
12. Draw the I/O interface timing diagram for read – write read sequence of operation.
13. What are interrupts? How interrupts are handled by C54xx DSP Processors.
14. Design a data memory system with address range 000800h – 000fffh for a c5416processor using 2kx8 SRAM memory chips. (**MAY-JUNE 10, 6m)**
15. Explain an interface between an A/D converter and the TMS320C54XX processor in the programmed I/O mode. . (**JUNE 12, 10m)**
16. Describe DMA with respect to TMS320C54XX processors.**(June/July 11, 10m)**
17. Drew the timing diagram for memory interface for read-read-write sequence of operation. Explain the purpose of each signal involved.**(June/July 11, 10m)**
18. Explain the memory interface block diagram for the TMS 320 C54xx processor.**(Dec 2010)**
19. Draw the I/O interface timing diagram for read – write read sequence of operation **(Dec2010)**
20. What are interrupts? How interrupts are handled by C54xx DSP Processors. **(Dec 2010,12)**
21. What are interrupts? What are the classes of interrupts available in the TMS320C54xx processor? **(JUNE/July 11, 8m)**

INTERFACING AND APPLICATIONS OF DSP PROCESSOR

❖ LEARNING OBJECTIVES

- Introduction,
- Synchronous Serial Interface,
- A CODEC Interface Circuit.
- DSP Based Bio-telemetry Receiver,
- A Speech Processing System,
- An Image Processing System

8.1 Introduction:

In the case of parallel peripheral interface, the data word will be transferred with all the bits together. In addition to parallel peripheral interface, there is a need for interfacing serial peripherals. DSP has provision of interfacing serial devices too.

8.2 Synchronous Serial Interface:

- There are certain I/O devices which handle transfer of one bit at a time. Such devices are referred to as serial I/O devices or peripherals.
- Communication with serial peripherals can be synchronous, with processor clock as reference or it can be asynchronous.
- Synchronous serial interface (SSI) makes communication a fast serial communication and asynchronous mode of communication is slow serial communication. However, in comparison with parallel peripheral interface, the SSI is slow. The time taken depends on the number of bits in the data word.

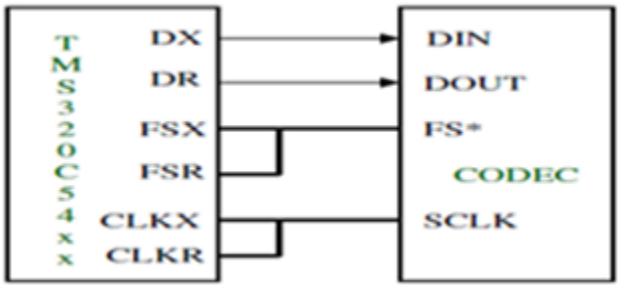


FIG: 8.1 SSI between DSP and CODEC

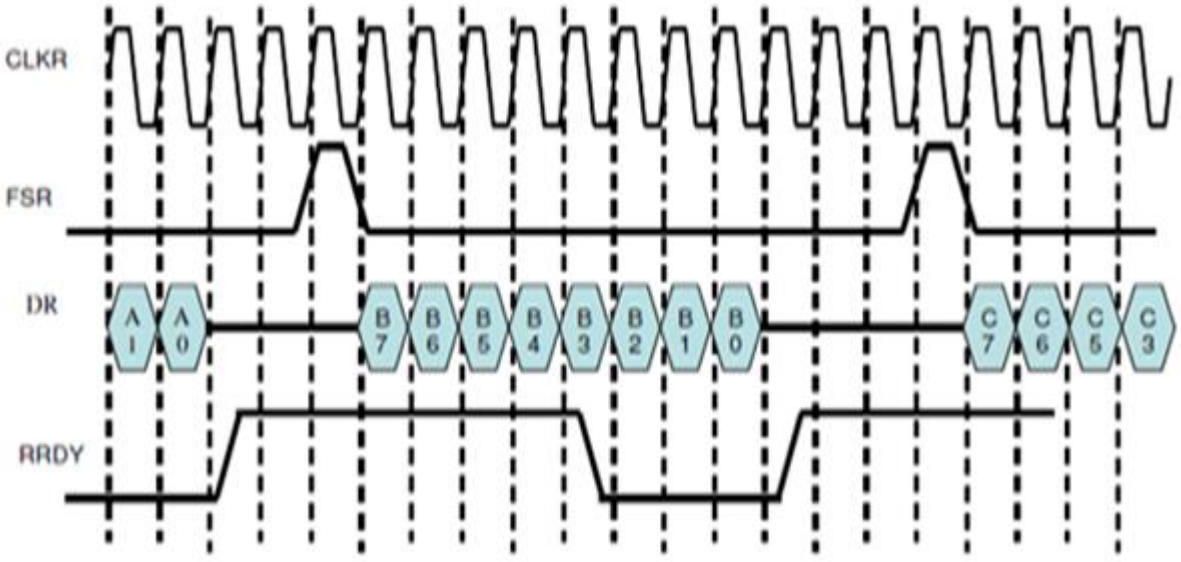


FIG: 8.2 receiving timing for SSI

- As shown, the receiving or transmit activity is initiated at the rising edge of clock, CLKR/ CLKX.
- Reception / Transfer start after FSR / FSX remains high for one clock cycle.
- RRDY / XRDY is initially high, goes LOW to HIGH after the completion of data transfer.
- Each transfer of bit requires one clock cycle. Thus, time required to transfer / receive data word depends on the number of bits in the data word.
- An example of data word of 8 bits is shown in the fig. 8.2 and fig. 8.3.

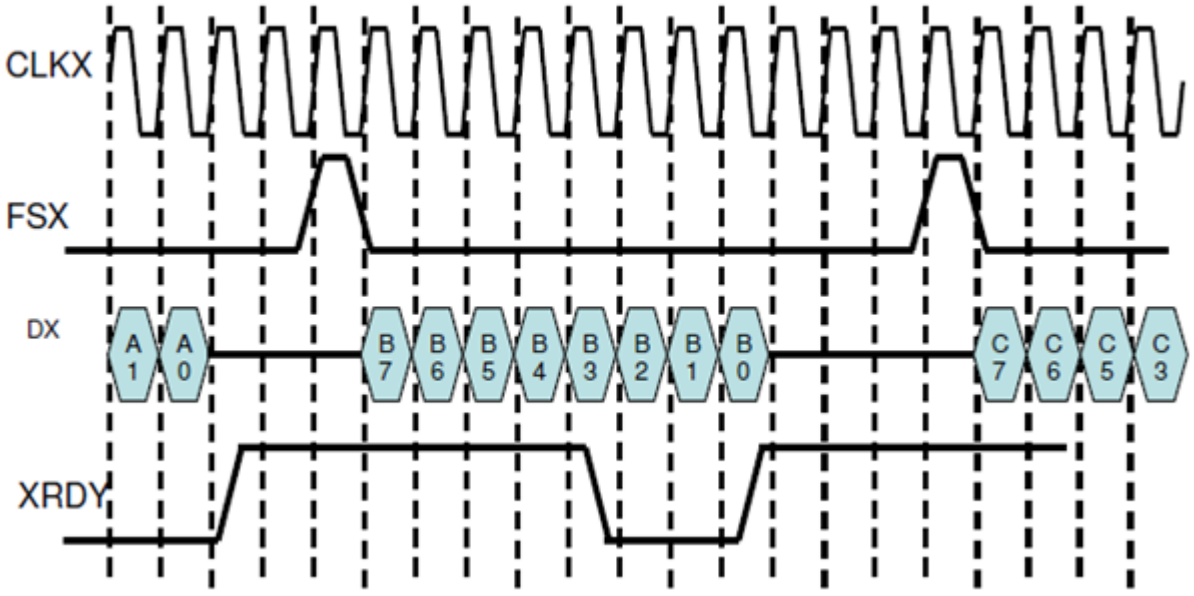


FIG: 8.3 Transmit timing for SSI

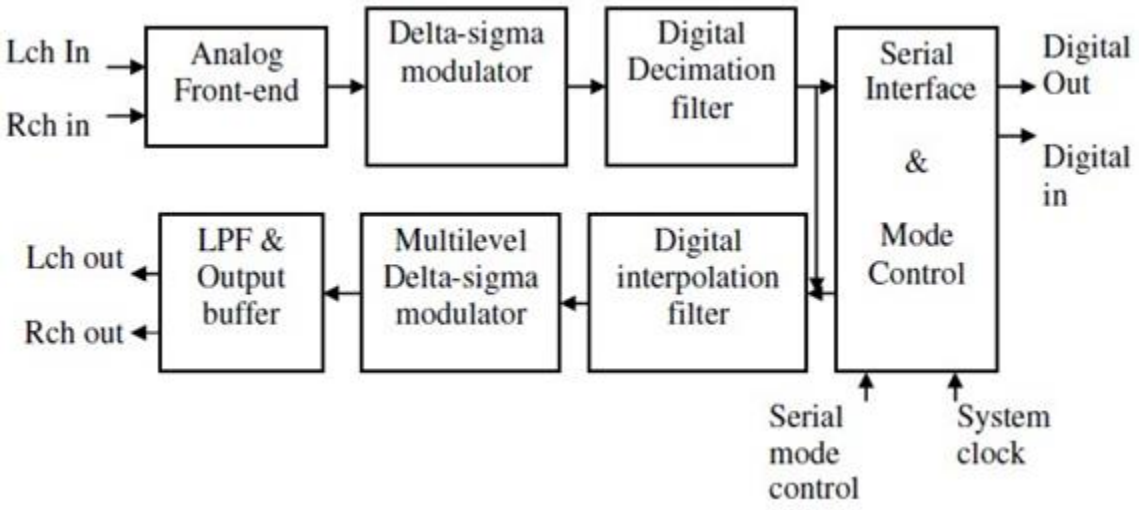


FIG: 8.4 Block diagram of PCM3002 CODEC

- Fig. 8.4 shows the block diagram of PCM3002 CODEC.
- Analog front end samples signal at 64X over sampling rate. It eliminates need for sample-and-hold circuit and simplifies need for anti aliasing filter.
- ADC is based on Delta-sigma modulator to convert analog signal to digital form.
- Decimation filter reduces the sampling rate and thus processing does not need high speed devices.
- DAC is Delta-sigma modulator, converts digital signal to analog signal.
- Interpolation increases the sampling rate back to original value.
- LPF smoothens the analog reconstructed signal by removing high frequency components.
- The Serial Interface monitors serial data transfer. It accepts built-in ADC output and converts to serial data and transmits the same on DOUT.
- It also accepts serial data on DIN & gives the same to DAC. The serial interface works in synchronization with BCLKIN & LRCIN.
- The Mode Control initializes the serial data transfer. It sets all the desired modes, the number of bits and the mode Control Signals, MD, MC and ML.
- MD carries Mode Word. MC is the mode Clock Signal. MD to be loaded is sent with reference to this clock.
- ML is the mode Load Signal. It defines start and end of latching bits into CODEC device.

- Figure 8.5 shows interfacing of PCM3002 to DSP in DSK. DSP is connected to PCM3002 through McBSP2. The same port can be connected to HPI.
- Mux selects one among these two based on CPLD signal. CPLD in Interface also provides system clock for DSP and for CODEC, Mode control signals for CODEC.

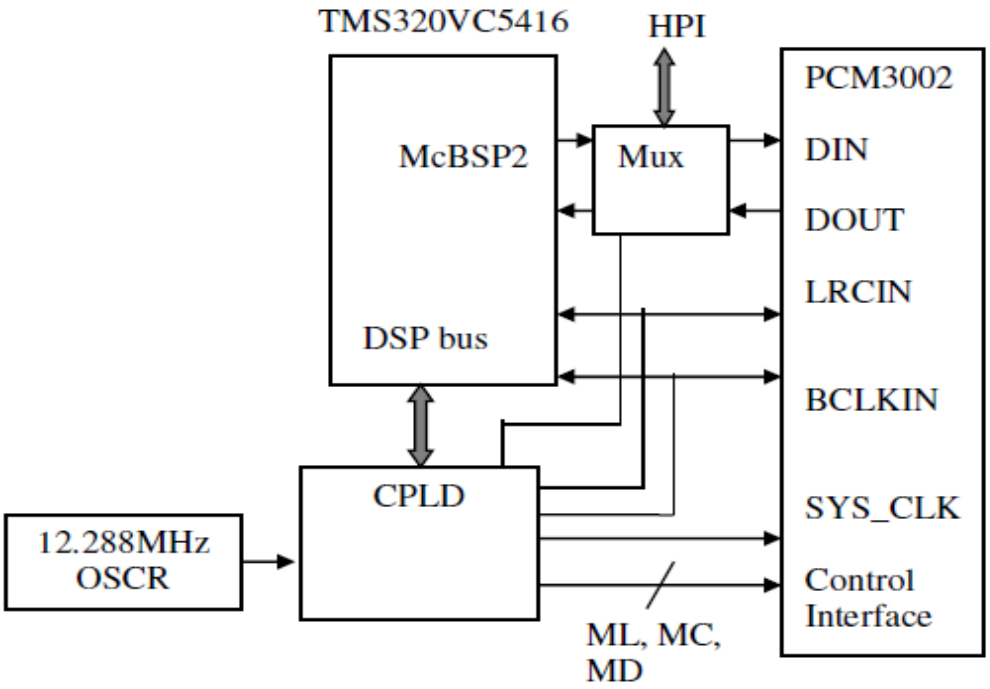


Fig. 8.5: PCM3003 Interface to DSP in DSK

- CPLD generates BCLKIN and LRCIN signals required for serial interface.
- PCM3002 CODEC handles data size of 16 / 20 bits. It has 64x over-sampling, delta-sigma ADC & DAC. It has two channels, called left and right.
- The CODEC is programmable for digital de-emphasis, digital attenuation, soft mute, digital loop back, power-down mode.
- System clock, SYSCLK of CODEC can be 256fs, 384fs or 512fs. Internal clock is always 256fs for converters, digital filters.
- DIN, DOUT are the single line data lines to carry the data into the CODEC and from CODEC.
- Another signal BCLKIN is data bit clock, the default value of which is CODEC SYSCLK / 4. LRCIN is frame sync signal for Left and Right Channels.
- The frequency of this signal is same as the sampling frequency. The default divide factor can be 2, 4, 6 and 8. Thus, sampling rate is minimum of 6 KHz and maximum of 48 KHz

Problem 8.1: A PCM3002 is programmed for the 12 KHz sampling rate. Determine the divisor N that should be written to the CPLD of the DSK and the various clock frequencies for the set up.

Solution: CPLD input Clock=12.288MHz (known) Sampling rate fs=CODEC_SYSCLK / 256 =12KHz (given) CPLD output clock, CODEC_SYSCLK =12.288 x 106 / N Thus, CODEC_SYSCLK =256 x 12 KHz

& N=12.288 x 106/(256 x 12 x 103)= 4

Problem 8.3: Frame Sync is generated by dividing the 8.192MHz clock by 256 for the serial communication. Determine the sampling rate and the time a 16 bit sample takes when transmitted on the data line.

Solution: LRCIN, Frame Sync = 8.192x106/256 =32 KHz Sampling rate fs= frequency of LRCIN=32 KHz

BCLKIN, Bit clock rate=CODEC_SYSCLK / 4=8.192x106/4=2.048MHz

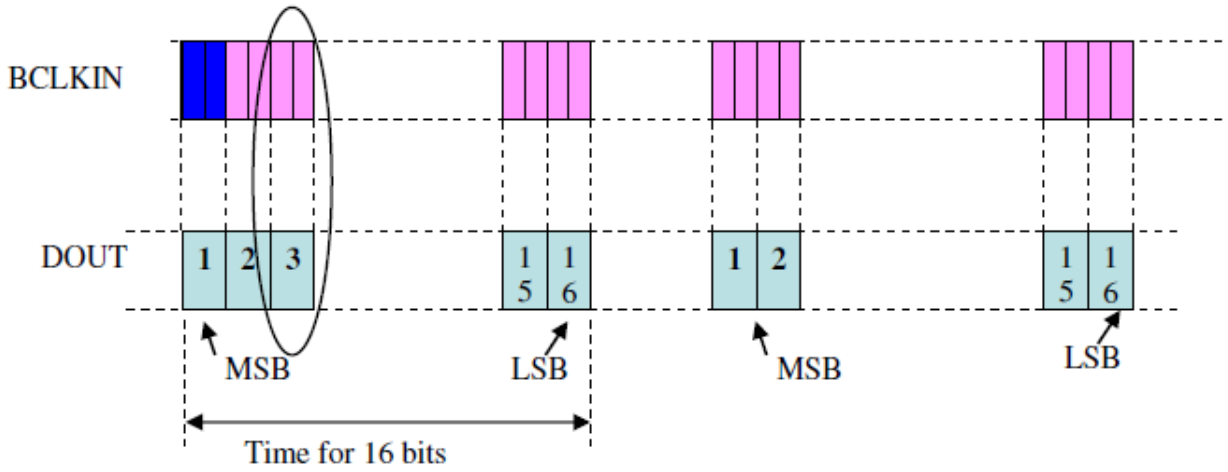


Fig. P8.3: Time for Data Transmission

LRCIN, Frame Sync = 8.192x10^6/256 =32 KHz Sampling rate fs= frequency of LRCIN=32 KHz

BCLKIN, Bit clock rate=CODEC_SYSCLK / 4=8.192x10^6/4=2.048MHz Bit clock period= 1/2.048x10^6 =0.488x10^-6s

Time for transmitting 16 bits =0.488x10^-6x16 =7.8125x10^-6s (refer fig. P8.3)

The CODEC PCM3002 supports four data formats as listed in table 8.1. The four data formats depend on the number of bits in the data word, if the data is right justified or left justified with respect to LRCIN and if it is I2S (**I**ntegrated **I**nter-chip **S**ound) format.

Table 8.1: Data formats of CODEC		
Format	DAC	ADC
Format 0	16 bit, MSB first, right justified	16 bit, MSB first, left justified
Format 1	20 bit, MSB first, right justified	20 bit, MSB first, left justified
Format 2	20 bit, MSB first, left justified	20 bit, MSB first, left justified
Format 3	20 bit, MSB first, I2S	20 bit, MSB first, I2S

Figure 8.6 and fig. 8.7 depicts the data transaction for CODEC PCM3002. As shown in fig. 8.6, DIN (/ DOUT) carries the data. BCLKIN is the reference for transfer. When LRCIN is high, left channel inputs (/ outputs) the data and when LRCIN is low, right channel inputs (/ outputs) the data. The data bits at the end (/ beginning) of the LRCIN thus Right (/ left) justified.

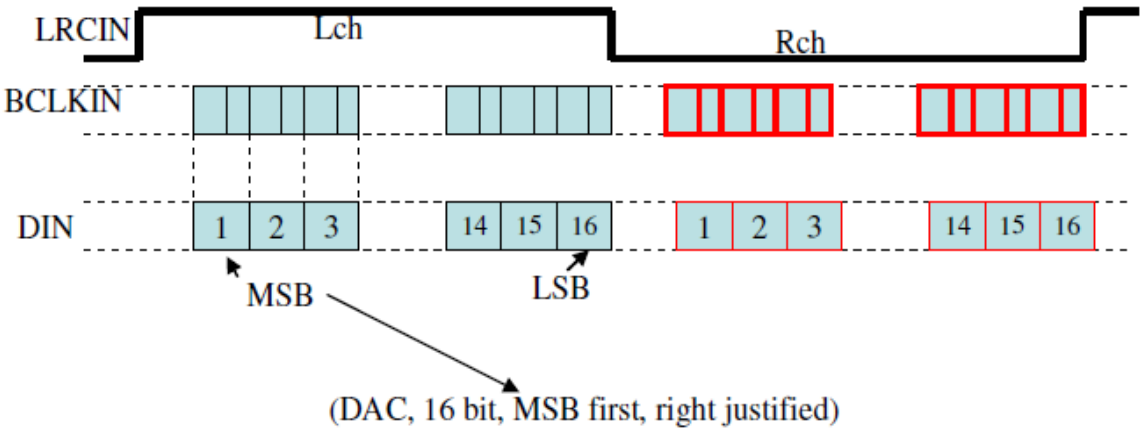


Fig. 8.6: Data Formats for PCM3002

Another data format handled by PCM3002 is I2S (Integrated Inter-chip Sound). It is used for transferring PCM between CD transport & DAC in CD player. LRCIN is low for left channel and high for right channel in this mode of transfer. During the first BCKIN, there is no transmission by ADC. During 2nd BCKIN onwards, there is transmission with MSB first and LSB last. Left channel data is handled first followed by right channel data.

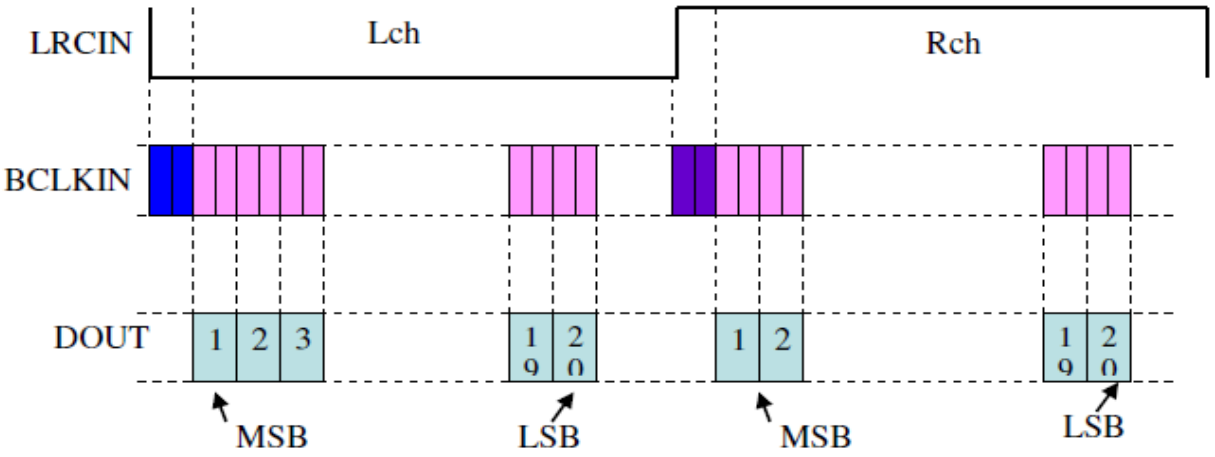


Fig. 8.7: ADC 20 bit, MSB first, I^2S format

8.3 DSP Based Bio-telemetry Receiver:

- Biotelemetry involves transfer of physiological information from one remote place to another for the purpose of obtaining experts opinion. The receiver uses radio Frequency links.
- The schematic diagram of biotelemetry receiver is shown in fig.8.8.
- The biological signals may be single dimensional signals such as ECG and EEG or two dimensional signals such as an image, i.e., X-ray. Signal can even be multi dimensional signal i.e., 3D picture.
- The signals at source are encoded, modulated and transmitted. The signals at destination are decoded, demodulated and analyzed.

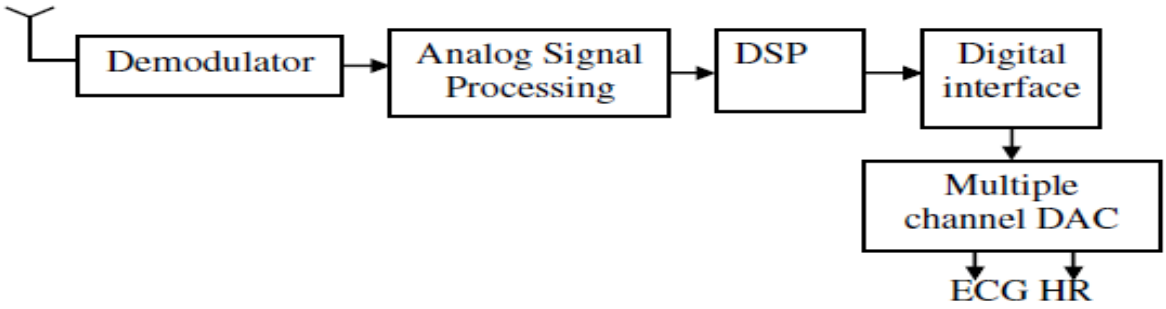
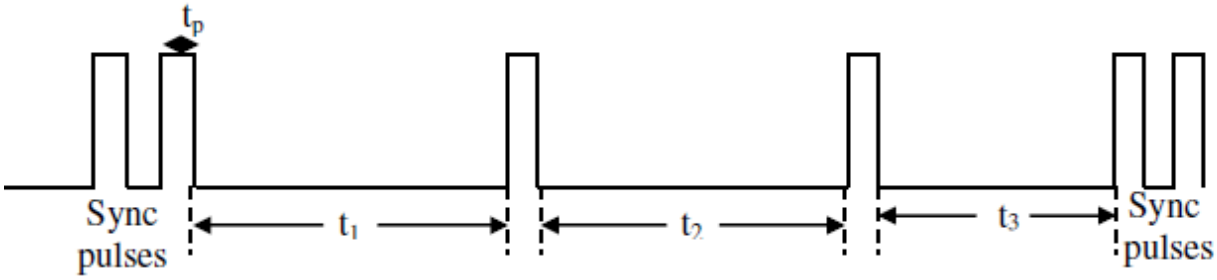


Fig. 8.8: Bio-telemetry Receiver

- An example of processing ECG signal is considered. The scheme involves modulation of ECG signal by employing Pulse Position Modulation (PPM).
- At the receiving end, it is demodulated. This is followed by determination of Heart beat Rate (HR). PPM Signal either encodes single or multiple signals. The principle of modulation being that the position of pulse decides the sample value.

- The PPM signal with two ECG signals encoded is shown in fig. 8.9. The transmission requires a sync signal which has 2 pulses of equal interval to mark beginning of a cycle.
- The sync pulses are followed by certain time gap based on the amplitude of the sample of 1st signal to be transmitted. At the end of this time interval there is another pulse.
- This is again followed by time gap based on the amplitude of the sample of the 2nd signal to be transmitted. After encoding all the samples, there is a compensation time gap followed by sync pulses to mark the beginning of next set of samples.
- Third signal may be encoded in either of the intervals of 1st or 2nd signal. With two signals encoded and the pulse width as tp, the total time duration is 5tp.



each pulse interval t1: pulse interval corresponding to sample value of 1st signal,t2: pulse interval corresponding to sample value of 2nd signal, t3: compensation time interval

Fig. 8.9: A PPM signal with two ECG signals

- Since the time gap between the pulses represent the sample value, at the receiving end the time gap has to be measured and the value so obtained has to be translated to sample value.
- The scheme for decoding is shown in fig. 8.10. DSP Internal Timer employed. The pulses in PPM generate interrupt signals for DSP. The interrupt start / terminate the timer.
- The count in the timer is equivalent to the sample value that has been encoded. Thus, ADC is avoided while decoding the PPM signal.

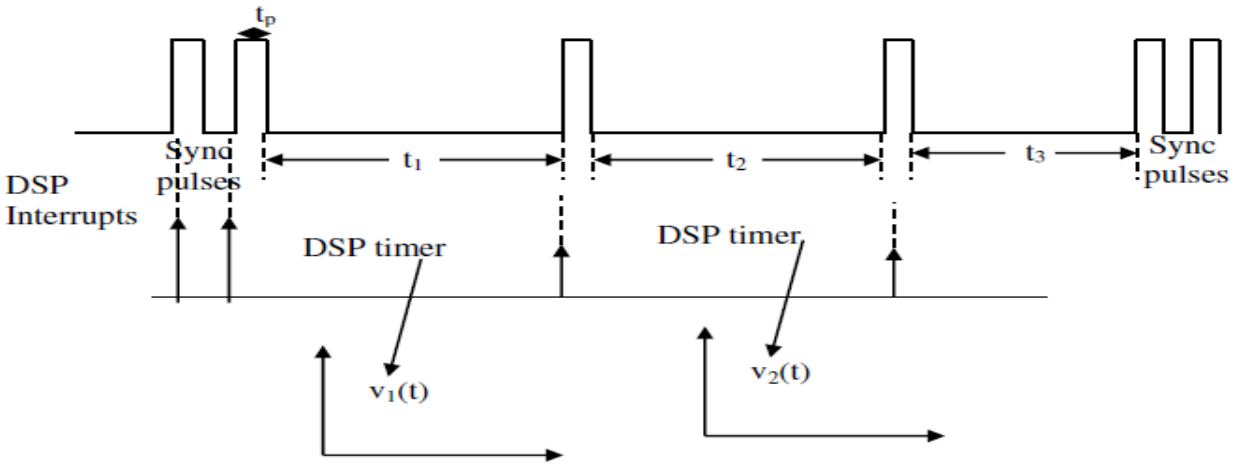


Fig. 8.10: Decoding PPM signal with two ECG signals

- A DSP based PPM signal decoding is shown in fig. 8.11. PPM signal interface generates the interrupt for DSP.
- DSP entertains the interrupt and starts a timer. When it receives another interrupt, it stops the timer and the count is treated as the digital equivalent of the sample value. The process repeats.
- Dual DAC converts two signals encoded into analog signals. And heart rate is determined referring to the ECG obtained by decoding

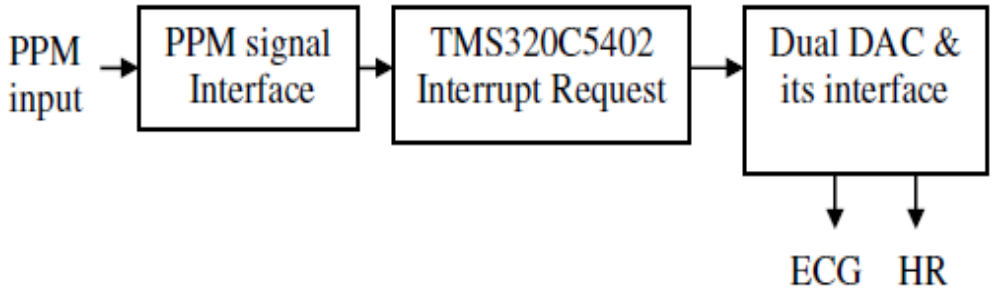


Fig. 8.11: DSP based biotelemetry Receiver Implementation

- Heart Rate (HR) is a measure of time interval between QRS complexes in ECG signal. QRS complex in ECG is an important segment representing the heart beat.
- There is periodicity in its appearance indicating the heart rate. The algorithm is based on 1st and 2nd order absolute derivatives of the ECG signal. Since absolute value of derivative is taken, the filter will be a nonlinear filtering.

Let the 1st order derivative be $y1(n) = |x(n) - x(n-1)|$

And let the 2nd order derivative be $y2(n) = |x(n-2) - 2x(n-1) + x(n)|$

The 1st order derivative is obtained as the difference between the two adjacent samples, the present sample and the previous sample. In a similar way, 2nd order derivative is obtained by finding the derivative of 1st order derivative. The y1(n) and y2(n) are summed. & $y3(n) = y1(n) + y2(n)$

High frequency components are removed from y3(n) by passing the same through a LPF to get $y4(n) = \alpha(y3(n) - y4(n-1)) + y4(n-1)$

Mean of half of peak amplitudes is determined, which is threshold for detection of QRS complex. QRS interval is then the time interval between two such peaks. Time Interval between two peaks is determined using internal timer of DSP. Heart Rate, heart beat perminute is computed using the relation HR=Sampling rate x 60 / QRS interval. The signals at various stages are shown in fig. 8.12.

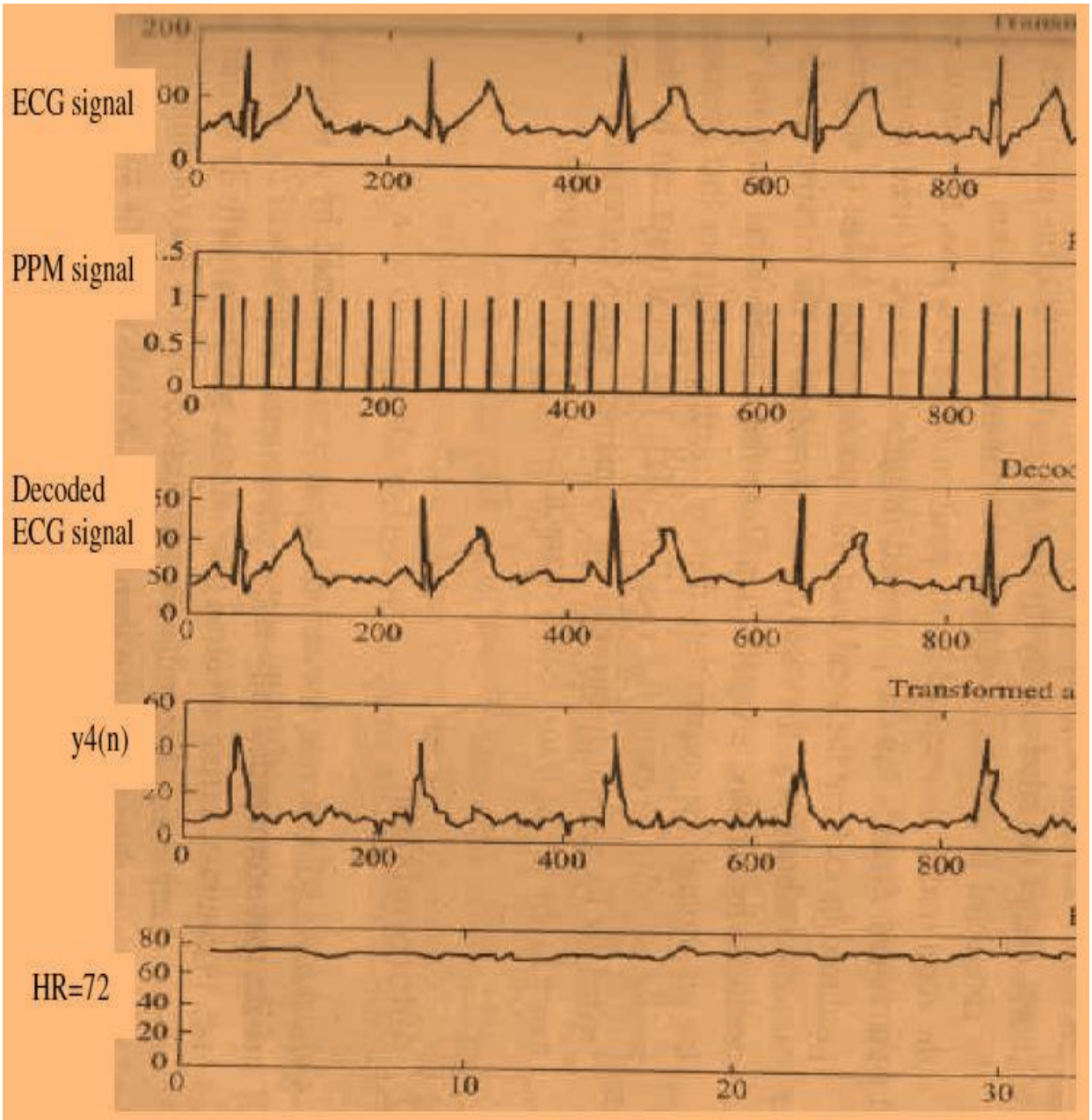


Fig. 8.12: Signals in determination of HR

8.4 A Speech Processing System:

- The purpose of speech processing is for analysis, transmission or reception as in the case of radio / TV / phone, denoising, compression and so on.
- There are various applications of speech processing which include identification and verification of speaker, speech synthesis, voice to text conversion and vice versa and so on.
- A speech processing system has a vocoder, a voice coding / decoding circuit. Schematic of speech production is shown in fig. 8.13.
- The vocal tract has vocal cord at one end and mouth at the other end. The shape of the vocal tract depends on position of lips, jaws, tongue and the velum. It decides the sound that is produced.
- There is another tract, nasal tract. Movement of velum connects or disconnects nasal tract. The overall voice that sounds depends on both, the vocal tract and nasal tract.
- Two types of speech are voiced sound and unvoiced sound. Vocal tract is excited with quasi periodic pulses of air pressure caused by vibration of vocal cords resulting in voiced sound.
- Unvoiced sound is produced by forcing air through the constriction, formed somewhere in the vocal tract and creating turbulence that produces source of noise to excite the vocal tract.

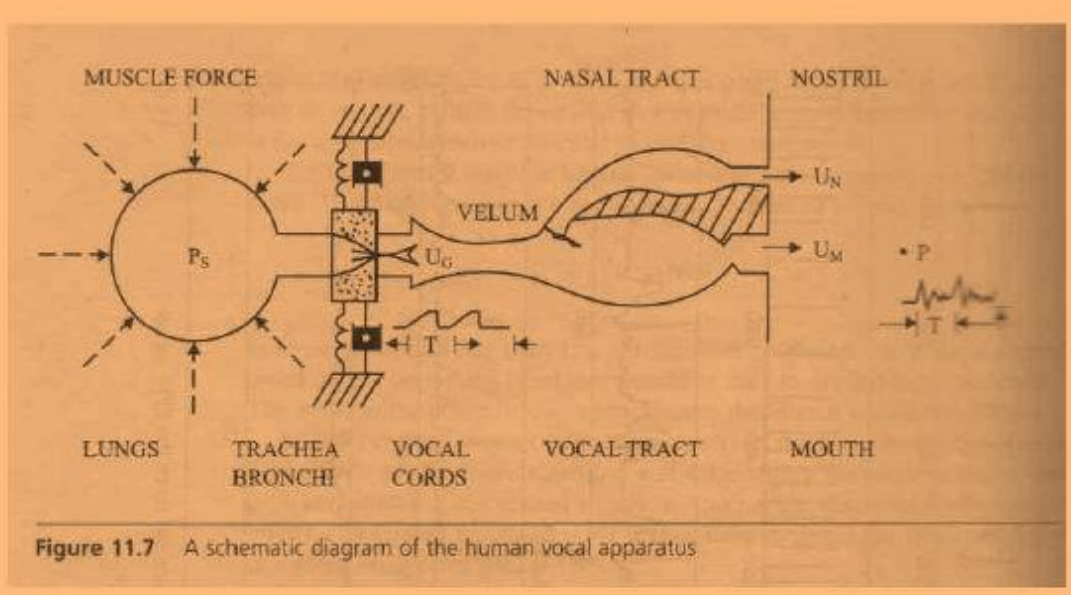


Fig. 8.13: Speech Production Mechanism

- By the understanding of speech production mechanism, a speech production model representing the same is shown in fig. 8.14. Pulse train generator generates periodic pulse train. Thus it represents the voiced speech signal.
- Noise generator represents unvoiced speech. Vocal tract system is supplied either with periodic pulse train or noise. The final output is the synthesized speech signal.

- Sequence of peaks occurs periodically in voiced speech and it is the fundamental frequency of speech. The fundamental frequency of speech differs from person to person and hence sound of speech differs from person to person.

- Speech is a non stationary signal. However, it can be considered to be relatively stationary in the intervals of 20ms. Fundamental frequency of speech can be determined by autocorrelation method. In other words, it is a method of determination of pitch period.

- Periodicity in autocorrelation is because of the fundamental frequency of speech.

- A three level clipping scheme is discussed here to measure the fundamental frequency of speech. The block diagram for the same is shown in fig. 8.15.

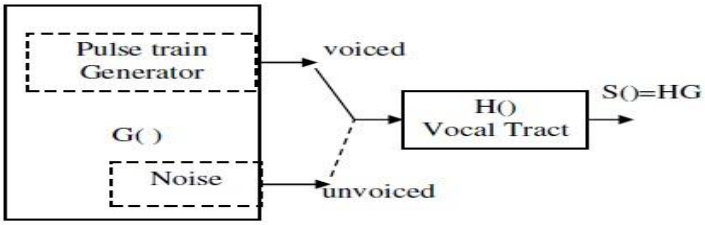


Fig. 8.14: Speech Production Model

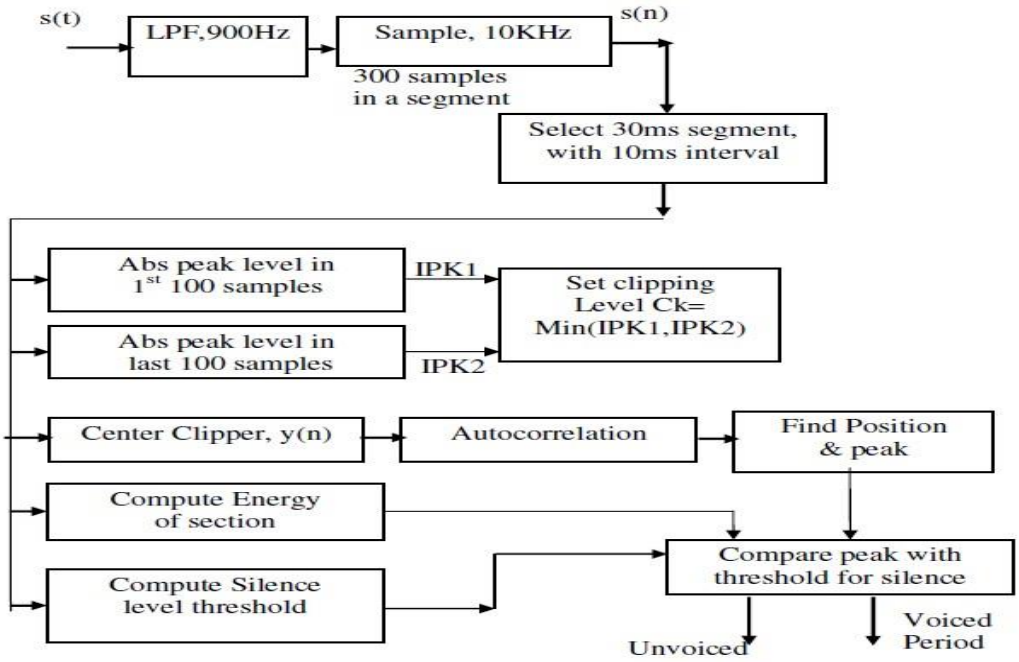


Fig. 8.15: Block Diagram of Clipping Autocorrelation Pitch Detector

- The speech signal s(t) is filtered to retain frequencies up to 900Hz and sampled using ADC to get s(n).

- The sampled signal is processed by dividing it into set of samples of 30ms duration with 20ms overlap of the windows. The same is shown in fig. 8.16

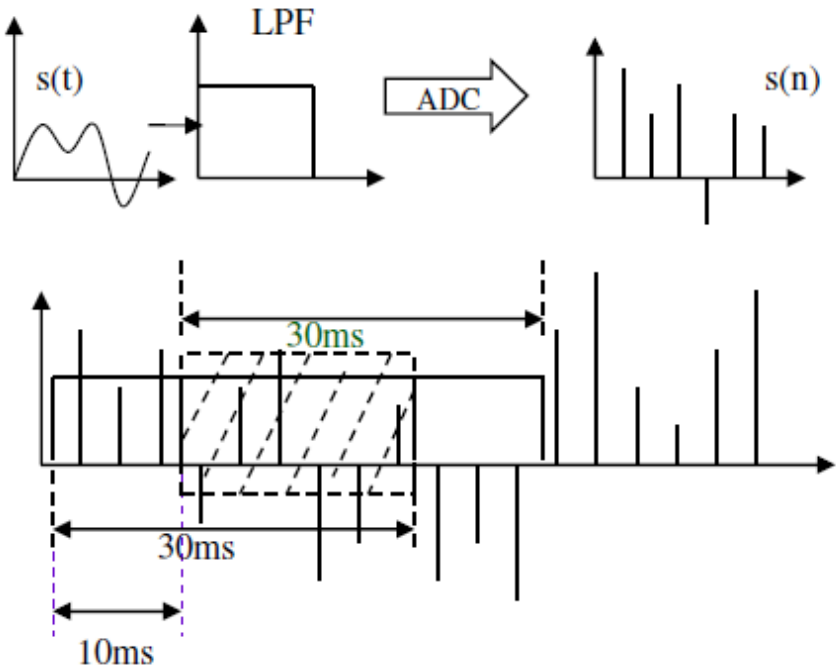


Fig. 8.16: LPF, ADC and windowing

- A threshold is set for three level clipping by computing minimum of average of absolute values of 1st 100 samples and last 100 samples. The scheme is shown in fig. 8.17.

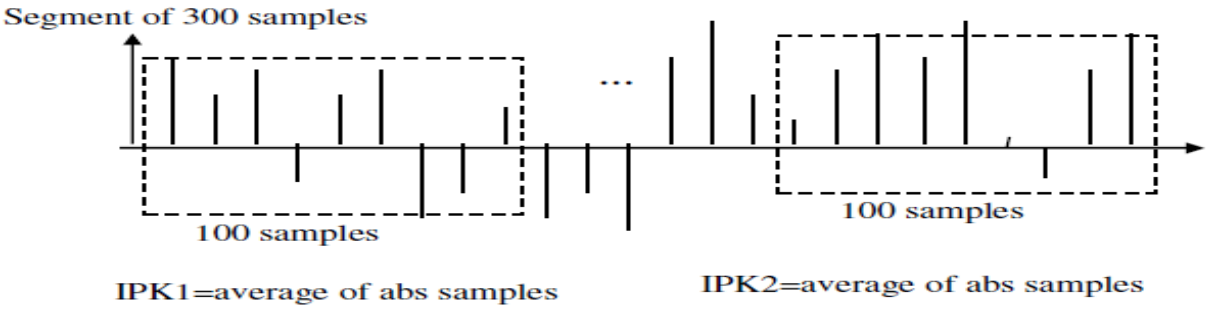


Fig. 8.17: Setting threshold for Clipping

- The transfer characteristics of three level clipping circuit is shown in fig. 8.18. If the sample value is greater than +CL, the output y(n) of the clipper is set to 1.

- If the sample value is more negative CL, the output y(n) of the clipper is set to -1. If the sample value is between –CL and +CL, the output y(n) of the clipper is set to 0.

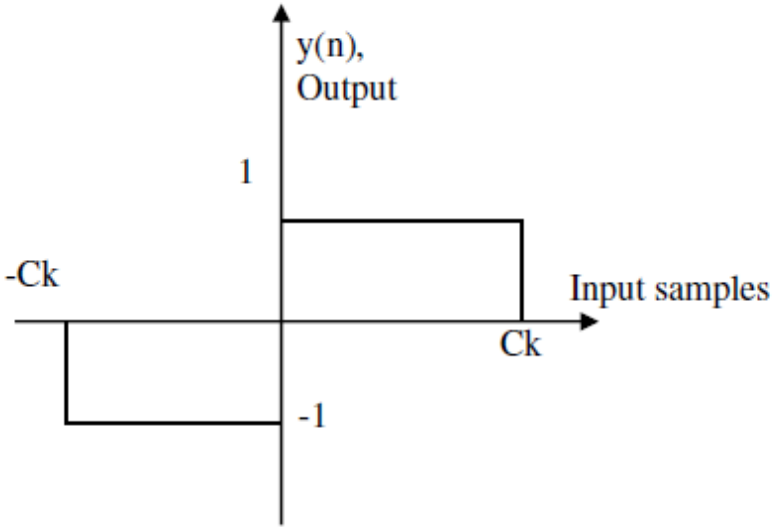


Fig. 8.18: Center Clipper

- The autocorrelation of y(n) is computed which will be 0, 1 or -1 as defined by eq (1). The largest peak in autocorrelation is found and the peak value is compared to a fixed threshold.
- If the peak value is below threshold, the segment of s(n) is classified as unvoiced segment. If the peak value is above threshold, the segment of s(n) is classified as voiced segment. The functioning of autocorrelation is shown in fig. 8.19.

$$R_n(k) = \sum_{m=0}^{N-1-k} y(n+m)\,y(n+m+k)$$

$$y(n+m)\,y(n+m+k) = \begin{cases} 0 \;\; if \;\; y(n+m) = 0 \;\; or \;\; y(n+m+k) = 0 \\ +1 \;\; if \;\; y(n+m) = y(n+m+k) \\ -1 \;\; if \;\; y(n+m) \neq y(n+m+k) \end{cases} \qquad (1)$$

- As shown in fig. 8.19, A is a sample sequence y(n). B is a window of samples of length N and it is compared with the N samples of y(n). There is maximum match. As the window is moved further, say to a position C the match reduces. When window is moved further say to a position D, again there is maximum match. Thus, sequence y(n) is periodic. The period of repetition can be measured by locating the peaks and finding the time gap between them.

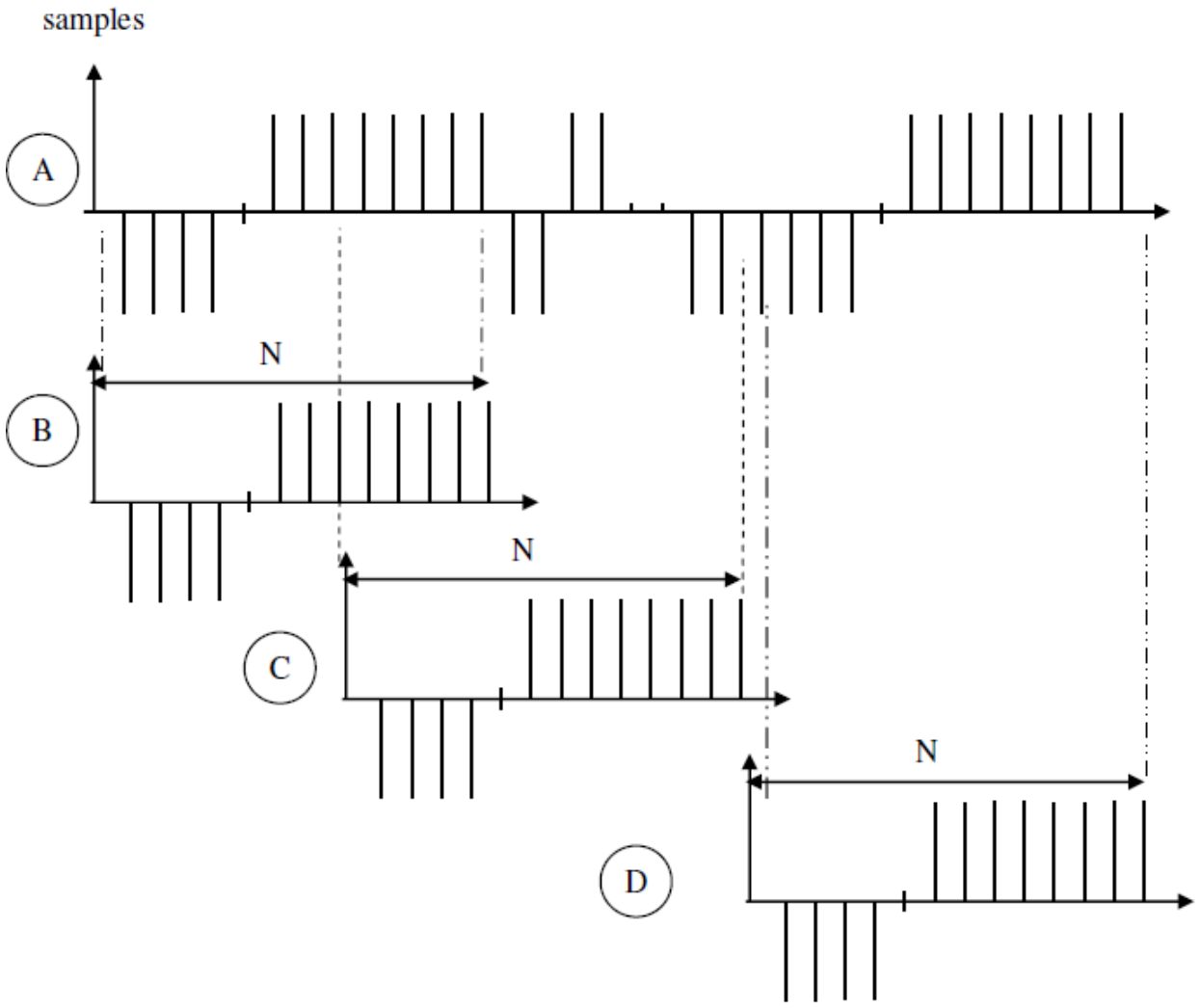


Fig. 8.19: Autocorrelation functioning

8.5 An Image Processing System:

- In comparison with the ECG or speech signal considered so far, image has entirely different requirements.
- It is a two dimensional signal. It can be a color or gray image. A color image requires 3 matrices to be maintained for three primary colors-red, green and blue.
- A gray image requires only one matrix, maintaining the gray information of each pixel (picture cell).
- Image is a signal with large amount of data. Of the many processing, enhancement, restoration, etc., image compression is one important processing because of the large amount of data in image. To reduce the storage requirement and also to reduce the time and band width required to transmit the

image, it has to be compressed.

- Data compression of the order of factor 50 is sometimes preferred. JPEG, a standard for image compression employs loss compression technique. It is based on discrete cosine transform (DCT).
- Transform domain compression separates the image signal into low frequency components and high frequency components. Low frequency components are retains because they represent major variations. High frequency components are ignored because they represent minute variations and our eye is not sensitive to minute variations.
- Image is divided into blocks of 8 x 8. DCT is applied to each block. Low frequency coefficients are of higher value and hence they are retained.
- The amount of high frequency components to be retained is decided by the desirable quality of reconstructed image. Forward DCT is given by eq (2).

$$f_{v,u} = \frac{1}{4} c_v c_u \sum_{x=0}^{7} \sum_{y=0}^{7} f_{x,y} \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \qquad (2)$$

- Since the coefficients values may vary with a large range, they are quantized. As already noted low frequency coefficients are significant and high frequency coefficients are insignificant, they are allotted varying number of bits. Significant coefficients are quantized precisely, with more bits and insignificant coefficients are quantized coarsely, with fewer bits.
- To achieve this, a quantization table as shown in fig. 8.20 is employed. The contents of Quantization Table indicate the step size for quantization. An entry as smaller value implies smaller step size, leading to more bits for the coefficients and vice versa.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig. 8.20: Matrix used for quantization & dequantization

- The quantized coefficients are coded using Huffman coding. It is a variable length coding Huffman Encoding. Shorter codes are allotted for frequently occurring long sequence of 1's & 0's.

- Decoding requires Huffman table and dequantization table. Inverse DCT is taken employing eq(3).
- The data blocks so obtained are combined to form complete image. The schematic of encoding and decoding is shown in fig. 8.21

$$f_{x,y} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} c_u c_v f_{u,v} \cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right) \qquad \text{eq(3)}$$

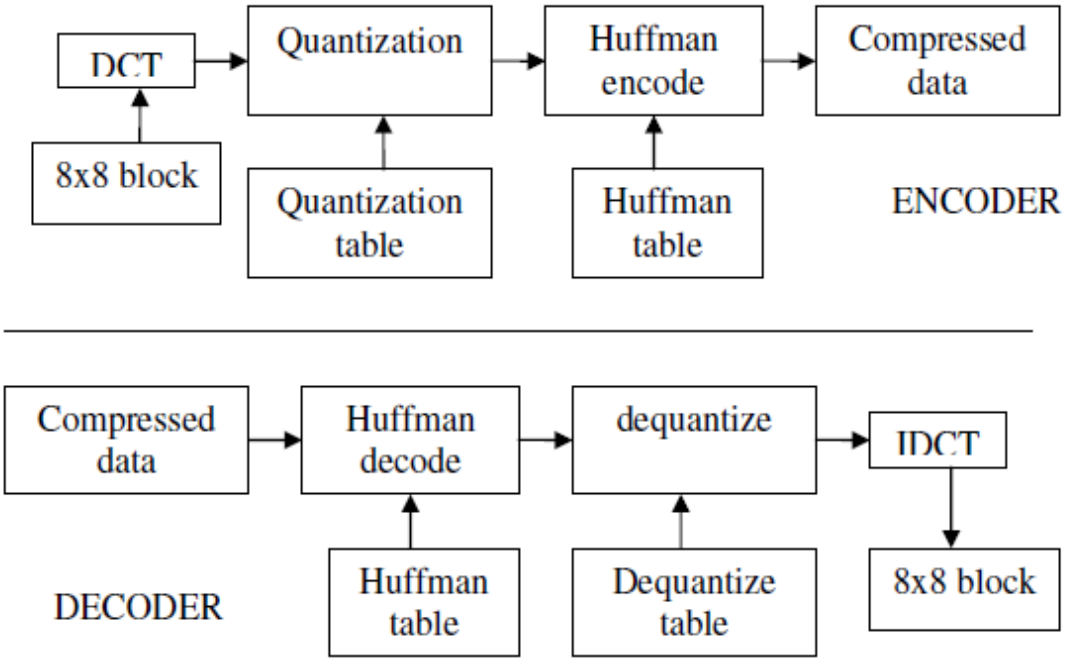


Fig. 8.21: JPEG Encoder & Decoder

ASSIGNMENT NO: 8 (Recommended Questions)

1. With the help of a block diagram, explain the image compression and reconstruction using JPEG encoder and decoder.[**JUNE/JULY-2011,5M.june-2014,10M.JUNE-2015,8M. DEC-2015,6M**]
2. Write a short note on i)DSP based telemetry receiver ii)Codec interface [**DEC-2011,10M**]
3. Write a pseudo algorithm heart rate (HR), using the digital signal processor.**[DEC-2015,6M]**
4. Explain briefly the building blocks of a PCM3002 CODECdevice. [**june-2012,6M.DEC-2012,8M.JUNE/JULY-2013,6M.JUNE-2014,10M. JUNE-2015,8M**]
5. What the help of block diagram, Explain DSP based biotelemetry receiver? [**june-2012,6M.DEC-2012,8M. JUNE/JULY-2013,6M.DEC-2014,10M**]
6. With the help of block diagram explain JPEG algorithm.[**DEC-2012,8M]**
7. Explain with the neat diagram the operation of pitch detector**.[june-2012,8M.DEC-2013,8M.**]
8. With a neat block diagram and timing diagram for both transmitter and receiver operation, explain the signal involved in synchronous serial interface.[**DEC-2013,12M**]
9. Explain with a neat diagram, the synchronous serial interface between the C54xx and a CODEC device. [**DEC-2014,10M.JUNE-2015,4M.JUNE-2016,6M**]
10. Explain the operation of pulse position modulation (PPM) to encode two biomedical signals. [**DEC-2015,6M. JUNE-2016,6M**]
11. Explain the memory interface block diagram for the TMS 320 C54xx processor.**(Dec 2010)**
12. Draw the I/O interface timing diagram for read – write read sequence of operation **(Dec 2010)**
13. Describe with a suitable diagram a digital model for production of speech signal.**[JUNE-2016,6M]**