SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

INTERNAL ASSESSMENT EXAMINATION – I

19SB405 – MICROPROCESORS AND ADVANCED MICROCONTROLLERS

ANSWER KEY

Part A

1. **How multi processors can be connected in 8086?**

   The 8086 processor is a single-chip microprocessor that does not support multiprocessing. However, multiple 8086 processors can be connected in a multi-processor system through a system bus.

2. **Write the instruction used to clear the carry Flag.**

   n 8086 assembly language, the instruction used to clear the carry flag is "CLC", which stands for "Clear Carry Flag".
   This instruction sets the carry flag to 0, indicating that there was no carry out of the most significant bit of the previous arithmetic operation

3.Define Assembler Directives

Assembler directives, also known as pseudo-operations, are commands used in assembly language programming to provide instructions to the assembler, rather than the processor.

4.List the advantages of DSP.

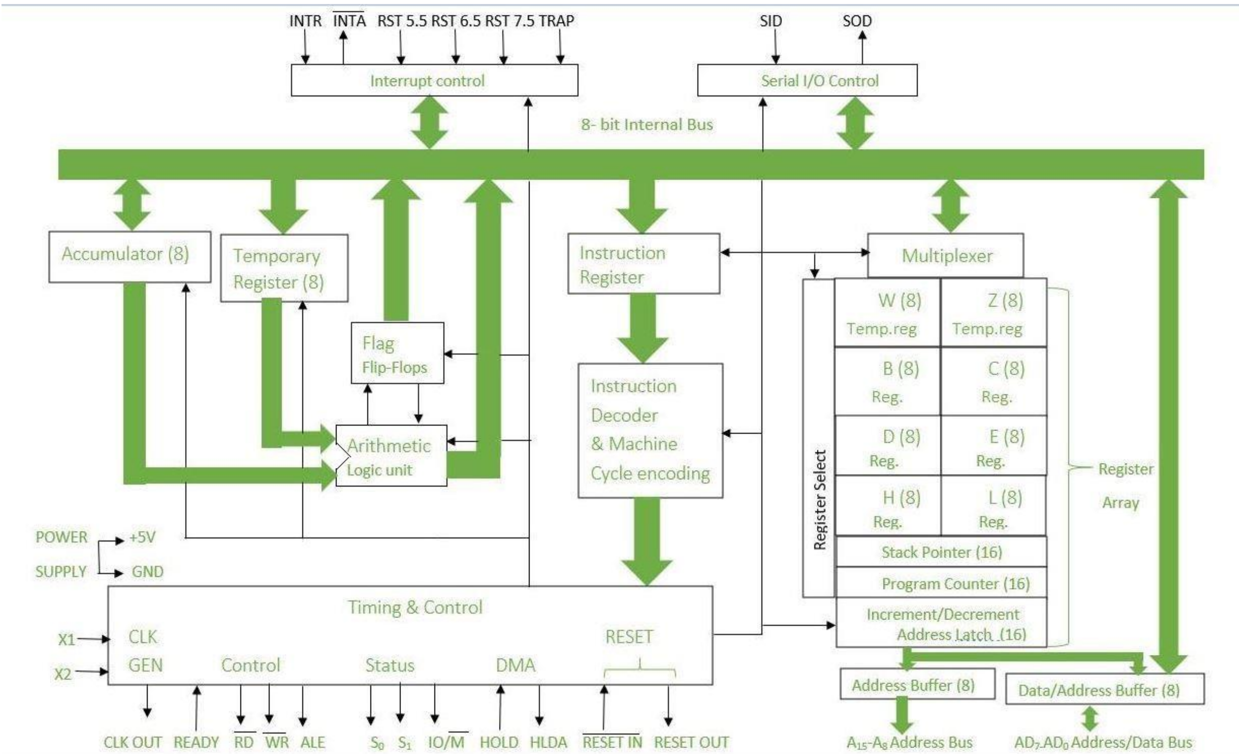
Flexibility
DSP algorithms can be modified
Accuracy
High precision
Less noise, distortion, and interference

5.Compare Floating and Fixed point processors.

Floating-point and fixed-point processors are two types of digital signal processors that differ in how they represent and process numerical data.
Here are some of the key differences between the two.

Part B

6(a) Describe the Architecture of 8085 with neat diagram



6(b) .Classify the instruction set of 8086 with examples.

The instruction set of the 8086 microprocessor can be classified into the following categories:

- Data transfer instructions: These instructions are used to move data between registers and memory locations. Examples include MOV, XCHG, PUSH, and POP.
- Arithmetic instructions: These instructions are used to perform arithmetic operations on data. Examples include ADD, SUB, INC, and DEC.
- Logical instructions: These instructions are used to perform logical operations on data. Examples include AND, OR, XOR, and NOT.
- Branching instructions: These instructions are used to alter the normal sequence of program execution. Examples include JMP, JZ, JC, and CALL.
- Flag control instructions: These instructions are used to modify the contents of the flag register. Examples include CLC, STC, and CMC.
- String instructions: These instructions are used to perform operations on strings of data. Examples include MOVS, CMPS, and SCAS.
- Input/output instructions: These instructions are used to transfer data between the microprocessor and external devices. Examples include IN and OUT.Miscellaneous instructions: These instructions perform other operations such as NOP, HLT, and WAIT.

7(a) Explain the addressing modes of 8086 with an example. (ii) Discuss string manipulation instruction

Addressing modes of 8086:

The 8086 microprocessor supports several addressing modes that determine how operands are accessed by the instruction. The addressing modes of the 8086 microprocessor are as follows:Immediate addressing mode: The operand is a constant data value that is specified in the instruction. For example, MOV AX, 1234H.

- Register addressing mode: The operand is a register. For example, MOV AX, BX.
- Direct addressing mode: The operand is a memory location whose address is
- Indirect addressing mode: The operand is a memory location whose address is specified indirectly through a register or an indexed addressing scheme. For example, MOV AX, [BX].
- Base-index addressing mode: The operand is a memory location whose address is calculated by adding a base register and an index register. For example, MOV AX, [BX+SI].
- Relative addressing mode: The operand is a memory location whose address is calculated relative to the current instruction pointer. For example, JMP SHORT LABEL.

(ii) String manipulation instruction:

The 8086 microprocessor supports several instructions for string manipulation. These instructions operate on a string of data and are used to perform operations such as copying, comparing, and searching strings. The string manipulation instructions of the 8086 microprocessor are as follows:

- MOVS (Move String): This instruction copies a byte or a word from the source string to the destination string and increments or decrements the source and destination pointers. For example, MOVS BX, [SI], MOVS [DI], AX.
- CMPS (Compare String): This instruction compares a byte or a word from the source string with a byte or a word from the destination string and sets the flags accordingly. For example, CMPS [SI], [DI].
- SCAS (Scan String): This instruction compares a byte or a word from the source string with the contents of the accumulator and sets the flags accordingly. For example, SCASB, SCASW.
- LODS (Load String): This instruction loads a byte or a word from the source string to the accumulator or a register and increments or decrements the source pointer. For example, LODS AL, LODS AX.
- STOS (Store String): This instruction stores a byte or a word from the accumulator or a register to the destination string and increments or decrements the destination pointer.

7(b)Analyze and conclude how the DSP is faster in manipulation of instructions than Microprocessor.

- Digital Signal Processors (DSPs) and microprocessors are both types of specialized microcontrollers used for processing digital signals, but DSPs are designed to perform complex mathematical operations on signals at a much faster rate than microprocessors.

- Specialized Architecture: DSPs are designed with specialized hardware and instruction sets optimized for performing signal processing operations such as multiplication, addition, and subtraction. In contrast, microprocessors have a more general-purpose architecture, designed to handle a wide range of tasks. This specialized architecture of DSPs allows them to perform specific operations much faster than a general-purpose microprocessor.
- Parallelism: DSPs are designed to perform multiple operations in parallel, which makes them much faster than microprocessors in handling signal processing tasks. DSPs use multiple functional units that can operate simultaneously, while microprocessors typically have a single arithmetic logic unit (ALU) that can only perform one operation at a time.
- Memory architecture: DSPs have a separate data and instruction memory, whereas microprocessors have a unified memory architecture. This separate memory architecture of DSPs allows for simultaneous access to both instruction and data memory, which improves the efficiency of the DSP.
- Low power consumption: DSPs are designed to consume less power compared to microprocessors, which makes them ideal for portable or battery-powered devices that require signal processing capabilities.

- In conclusion, DSPs are faster than microprocessors in the manipulation of instructions because of their specialized architecture, parallelism, memory architecture, and low power consumption. They are specifically designed to handle complex signal processing operations at a much faster rate

Part - C

8(a). Create a program to perform Division, Multiplication and sorting of Numbers in Ascending and Descending order. Function to perform

Multiplication :

```
ASSUME CS:CODE,DS:CODE
        CODE SEGMENT
        ORG 1000H
        MOV SI,2000H
        MOV DI,3000H
        MOV AX,[SI]
        ADD SI,02H
        MOV BX,[SI]
        IMUL BX
        MOV [DI],AX
        ADD DI,02H
        MOV [DI],DX
        MOV AH,4CH
        INT 21H
        CODE ENDS
        END
```

Division :

```
ASSUME CS:CODE,DS:CODE
       CODE SEGMENT
       ORG 1000H
       MOV DI,2000H
       MOV AX,0FFFFH
       MOV BX,7777H
       DIV BX
       MOV [DI],AX
       MOV AH,4CH
       INT 21H
       CODE ENDS
       END
```

8(b)Analyze the architecture and working of DSP with neat sketch.

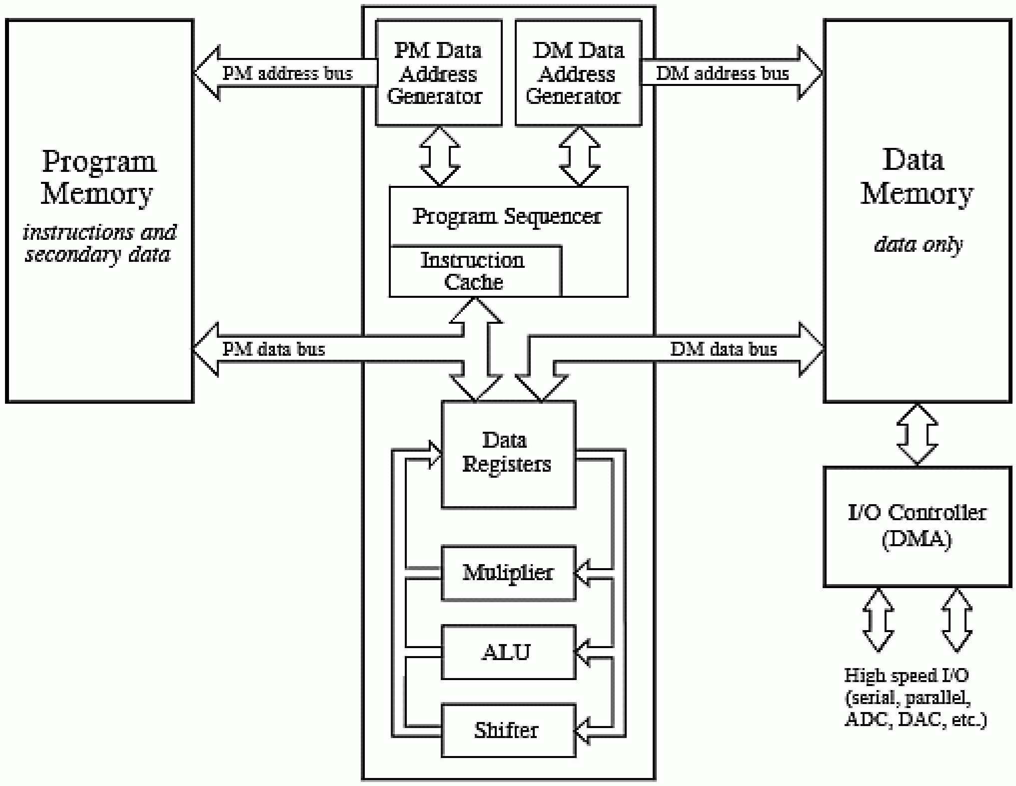


FIGURE 28-5
Typical DSP architecture. Digital Signal Processors are designed to implement tasks in parallel. This simplified diagram is of the Analog Devices SHARC DSP. Compare this architecture with the tasks needed to implement an FIR filter, as listed in Table 28-1. All of the steps within the loop can be executed in a single clock cycle.