full adder of the next higher bit and so on. The carry out of the most significant bit full adder is the output of the circuit. Logically it is the same as that of addition operation performed by us. We do pass the carry of lower digits addition to higher digits. The following Boolean function represents the output of this adder circuit:

$$O = X + Y + C_{in}$$

Please note that in Figure 10.15, the value of X is a direct input, but the value of Y is input through the multiplexer using the selection input S. In addition, the value of C_{in} is another input. The arithmetic micro-operations that can be implemented using Figure 10.15 are given in Figure 10.16.

S	C_{in}	Y	$O = X+Y+C_{in}$	Equivalent Micro-Operation	Micro-Operation Name
0	0	B	O = A + B	R ← R1 + R2	Add
0	1	B	O = A + B + 1	R ← R1 + R2 + 1	Add with carry
1	0	B′	O = A+B′	R ← R1 + R2′	Subtract with borrow
1	1	B′	O = A+ B′ + 1	R ← R1 + 2's complement of R2	Subtract

Figure 10.16: Arithmetic Micro-operations implemented using Figure 10.15

Let us refer to some of the cases of the Figure 10.16.

When S= 0, input line B is applied directly to the Y inputs of the full adder. Now,

If input carry C_{in}= 0, the output will be O = A + B
If input carry C_{in}= 1, the output will be O = A + B + 1.

If you choose S= 1, thenB′ forms the Y input to the full adder. So,
If C_{in}= 1, then output D = A + B′ + 1. This is called subtractmicro-operation. (Why?)

Reason: Please observe the following example, where A = 0111 and B=0110, then B′=1001. The sum will be calculated as:

```
  0111      (Value of A)
+ 1001      ( Complement of B)
-----------------------------------
1 0000 + (ignore the carry out bit and Add Carry in = 1)
= 0001
```

Thus, it is a subtract micro-operation.

If C_{in}= 0, then D = A + B′. This is called subtract with borrow micro-operation. (Why?). Let us look into the same addition as above:

```
    0111      (Value of A)
    1001      ( Complement of B)
1   0000 + (Carry in =0) = 0000
```

This operation, thus, is equivalent to:

O = A +B′
O = (A – 1) + (B′ + 1)
=> O = (A – 1) + 2's complement of B
=> O = A – (B+1) Thus, is the name subtract with borrow

Two special cases:
When S= 0, C_{in}=1 and Y input is ZERO.

The output O = A + 0 + C_{in} =>O = A + 1

This micro-operation is an increment micro-operation.
When S= 1,C_{in}=0 and input word Y has all 1's.

Output O = A + All (1s) + C_{in} => D = A – 1 (How? Let us explain with the help of the following example).

This is a decrement micro-operation.

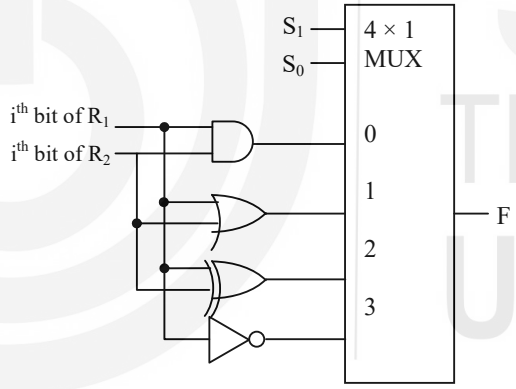
Example: Let us assume that the Register A is of 4 bits and contains the value 0101 and it is added to an all (1) value as:

```
  0101
+ 1111
------
1 0100
------
```

The 1 is carry out and is discarded. Thus, on addition with all (1's) the number has actually got decremented by one.

Implementation of Logic Micro-operations

In many computers only four logic micro-operations, viz. AND, OR, XOR and NOT logicmicro-operations, are implemented. The other logic micro-operations can be derived from these four micro-operations. Figure 10.17 shows one bit, which is the i^{th}bit stage of the four logic operations. Please note that the circuit consists of 4 gates and a 4 × 1 MUX. The i^{th} bits of Register R1 and R2 are passed through the circuit. On the basis of selection inputs S_0 and S_1 the desired micro-operation is obtained.



(a) Logic Diagram

S_1	S_0	Output	The Operation
0	0	$F = R_1 \wedge R_2$	AND Operation
0	1	$F = R_1 \vee R_2$	OR Operation
1	0	$F = R_1 \oplus R_2$	XOR Operation
1	1	$F = R_1'$	Complement of Register R_1

(b) Functional representation

Figure 10.17: Logic diagram of one stage of logic circuit

Arithmetic, Logic and Shift Unit

So, by now we have discussed how the arithmetic and logic micro-operations can be implemented individually. If we combine these two circuits along with shifting logic then we can have a possible simple structure of ALU. In effect ALU is a combinational circuit whose inputs are contents of specific registers. The ALU performs the desired micro-operation as determined by control signals on the input and places the results in an output or destination register. The whole operation of ALU can be performed in a single clock pulse, as it is a combinational circuit. The shift operation can be performed in a separate shift registers but sometimes it can be made as a part of overall ALU. More details on ALU can be studied from the further readings.

10.7 ARITHMETIC PROCESSORS

Arithmetic processorswere needed in the older computer processors to perform arithmetic processing, especially the floating-point arithmetic, as those processorsdid not had the required logic circuits to directly handle such processing. They were also called co-processor, as they were used as an additional processor of some main processor. However, in this era of ultra-large-scaleintegration and beyond, all the fixed point and floating-point computational capabilities are built in the processor. The concept relating to arithmetic processor is explained below.

A typical processor needs most of the control and data processing hardware for implementing non-arithmetic functions. As the hardware costs are directly related to chip area, a floating-point circuit being complex in nature is costly to implement. They need not be included in the instruction set of a processor. In such systems, floating-point operations were implemented by using software routines. This implementation of floating-point arithmetic is definitely slower than the hardware implementation. Now, the question is whether a processor can be constructed only for arithmetic operations. A processor, if devoted exclusively to arithmetic functions, can be used to implement a full range of arithmetic functions in the hardware at a relatively low cost. This can be done in a single Integrated Circuit. Thus, a special purpose arithmetic processor, for performing only the arithmetic operations, can be constructed. This processor physically may be separate yet can be utilized by the processor to execute complex arithmetic instructions. Please note in the absence of arithmetic processors, these instructions may be executed using the slower software routines by the processor itself. Thus, this auxiliary processor enhances the speed of execution of programs having a lot of complex arithmetic computations.

An arithmetic processor also helps in reducing program complexity, as it provides a richer instruction set for a machine. Some of the instructions that can be assigned to arithmetic processors can be related to the addition, subtraction, multiplication, and division of floating-point numbers, exponentiation, logarithms and other trigonometric functions.

How can this arithmetic processor be connected to the CPU?

If an arithmetic processor is treated as one of the Input / Output or peripheral units then it is termed as a peripheral processor. The CPU sends data and instructions to the peripheral processor, which performs the required operations on the data and communicates the results back to the CPU. A peripheral processor has several registers to communicate with the CPU. These registers may be addressed by the CPU as Input /Output register addresses. The CPU and peripheral processors are normally quite independent and communicate with each other by exchange of information using data transfer instructions. This type of connection is called loosely coupled.

If the arithmetic processor has a register and instruction set which can be considered an extension of the CPU registers and instruction set, then it is called a tightly coupled processor. Here the CPU reserves a special subset of code for arithmetic processor. In such a system the instructions meant for arithmetic processor are fetched by CPU and decoded jointly by CPU and the arithmetic processor, and finally executed by arithmetic processor. Thus, these processors can be considered a logical extension of the CPU. Such attached arithmetic processors were termed as co-processors.

These days floating point units are implemented as a part of the processor itself. More details on these can be found in further readings.

☞Check Your Progress 4

1. Explain the implementation of ALU.

……

……

……

Instruction fetch: fetching the

2. What is an Arithmetic Processor?

……

……

……

10.8 SUMMARY

This unit discusses the concept of instruction execution for a hypothetical machine with the help of micro-operations. It also describes very simplified view of implementation of micro-operations using combinational and sequential circuits. The idea is to give you a basic information about the implementation of a computer system based on its instruction set. The unit also discusses the concept of register transfer language for representing the micro-operations. The unit also defined the concept of Instruction Pipeline. The unit also discussed the hardware implementation of micro-operations. The unit shows a simple implementation of bus, which is the backbone for any register transfer operation. This is followed by a discussion on arithmetic circuit and micro-operation there on using full adder circuits. The logic micro-operation implementation has also been discussed. Finally, the unit also discussed the arithmetic processors.
You may refer to the further readings for more details on micro-operation concept and instruction cycle.

10.9 SOLUTIONS / ANSWERS

Check Your Progress 1

1. An address register is used to store memory address or can be used to compute memory address of instructions or operands.
2. To address 20 registers, you may require address field of length 5 bits, as $2^5 = 32$, thus, about 12 addresses are unused.
3. Independent set of conditional codes can help in parallel checking of conditions.
4. Yes. Several operating system allocate memory space for storing such information for later use.

Check Your Progress 2

1. Memory read operation requiresthe address of the location to be read. This address is first applied on the address BUS and the control unit enables memory read. Thus, the content of the addressed location is put in the data BUS. At the same time a data register is enabled to store the data on the data BUS. These operations can be represented as: Address BUS ←MAR ;DR ← Data BUS. The overall operation can be represented as: DR ← (MAR)
   In memory write operationthe memory address, where data is to be written, is applied on address BUS and the data that is to be stored/written is put on the data BUS. At the same time memory write operation is enabled by the control unit. This operation can be represented as: (MAR) ← DR

2. No, as multiplication operation will be implemented using addition and shift micro-operations.
3. (i) AND with R2 containing 0000 0000
   (ii) Initially XOR of R1 with R1 will clear R1, then perform OR with R2 having value 11001000

Check Your Progress 3

1. Indirect cycle is needed, when indirect memory addressing is used by an instruction. It converts an indirect address to a direct address. If an instruction is using register addressing scheme, then indirect cycle may be required only if instruction is using register indirect addressing. The indirect cycle in this case would require simple register transfer micro-operation.
2. Fetch cycle is primarily used for fetching instruction that is to be executed from the memory. The present-day machines may use instruction cache, instruction prefetch buffer etc., yet this instruction is needed to be moved to the processor, which may execute it. Thus, the nature of the fetch cycle may change but it may be required.
3. Interrupt cycle is responsible for acknowledging an interrupt. When an interrupt occurs in a computer, it is acknowledged, when the instruction execution gets completed. The processor checks, if an interrupt has occurred, if it is then an interrupt cycle is performed.

Check Your Progress 4

1. The implementation of ALU can be done with the help of combinational and sequential circuits. The combinational circuits perform the computations. The results of these computations is stored in sequential circuits. The internal register bus is used for input and output to ALU. Arithmetic circuit like adders may be used to perform arithmetic micro-operations, logic gates may be used to perform logic micro-operations and shift registers may be used to perform shift operations.
2. Arithmetic processor performs arithmetic computation. These may be support processors to a computer.