III-I

COMPUTER ORGANIZATION & ARCHITECTURE

REGISTER TRANSFER AND MICROOPERATIONS

Module 1 QB Solutions Handbook

Vishal | Syed Ikram | Pranav | Ujjwal

MODULE 1

PART A

1. Summarize the selection of address for control memory in the microprogrammed control unit

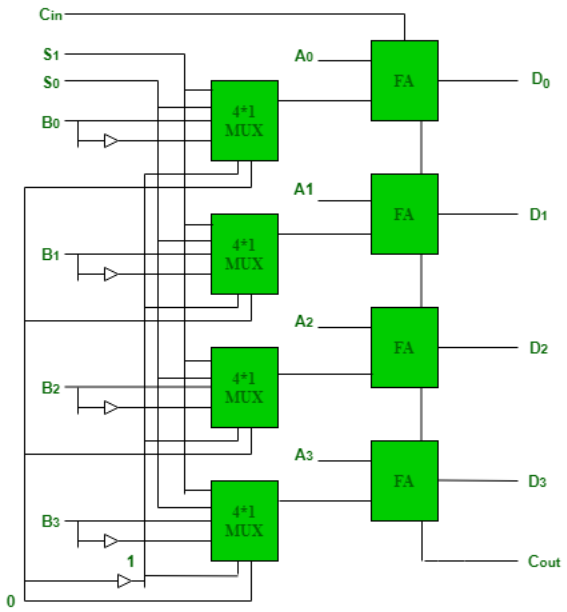
1. To execute an instruction, the sequencing logic unit issues a READ command to the control memory.
2. The word whose address is specified in the control address register is read into the control buffer register.
3. The content of the control buffer register generates control signals and next-address information for the sequencing logic unit.
4. The sequencing logic unit loads a new address into the control address register based on the next-address information from the control buffer register and the ALU flags.

2. Build the 4-bit arithmetic circuit for arithmetic micro-operations.

The basic component of an arithmetic circuit is the parallel adder. By controlling the data inputs to the adder, it is possible to obtain different types of arithmetic operations. It has four full-adder circuits that constitute the 4-bit adder and four multiplexers for choosing different operations. There are two 4-bit inputs A and B and a 4-bit output D. The four inputs from A go directly to the X

inputs of the binary adder. Each of the four inputs from B is connected to the data inputs of the multiplexers. The multiplexer's data inputs also receive the complement of B. The other two data inputs are connected to 0 and 1. 0 is a fixed voltage value (0 volts for TTL integrated circuits) and the 1 signal can be generated through an inverter whose input is 0. The four multiplexers are controlled by two selection inputs, S1 and S0. The input carries Cin goes to the carry input of the FA in the least significant position. The other carriers are connected from one stage to the next.

The output of the binary adder is calculated from the following arithmetic sum: D = A + Y + Cin where A is the 4-bit binary number at the X inputs and Y is the 4-bit binary number at the Y inputs of the binary adder. Cin is the input carry, which can be equal to 0 or 1. Note that the symbol + in the equation above denotes an arithmetic plus. By controlling the value of Y with the two selection inputs S1 and S0 and making Cin equal to 0 or 1, it is possible to generate the eight arithmetic micro-operations listed in Table 2-4.



3. Explain with an example how to multiply two unsigned binary numbers

Multiplication is different from addition in that multiplication of an n bit number by an m bit number results in an n+m bit number. Let's take a look at an example where n=m=4 and the result is 8 bits.

Decimal	Binary
10 x6 60	1010 x0110 0000 1010 1010 +0000 0111100

4. What is the need of a subroutine register in a control unit and discuss in detail?

A set of Instructions which are used repeatedly in a program can be referred to as Subroutine. Only one copy of this Instruction is stored in the memory. When a Subroutine is required it can be called many times during the Execution of a Particular program. Subroutines make programs shorter as well as easier to read and understand, because they break program code into smaller sections. You can test procedures or functions separately, rather than having to test the whole program. This makes programs easier to debug.

5. Demonstrate the need for some bits of current microinstruction to generate the address of the next microinstruction

The location of the next microinstruction may be the one next in sequence, or it may be located somewhere else in the control memory. For this reason, it is necessary to use some bits of the present microinstruction to control the generation of the address of the next microinstruction. The next address may also be a function of external input conditions. While the micro-operations are being executed, the next address is a computer in the next address generator circuit and then transferred into the control address register to read the next microinstruction. Thus a microinstruction contains bits for initiating micro-operations in the data processor part and bits that determine the address sequence for the control memory. The next address generator is sometimes called a microprogram sequencer, as it determines the address sequence that is read from control memory. The address of the next microinstruction can be specified in several ways, depending on the sequencer inputs. Typical functions of a microprogram sequencer are incrementing the control address register by one, loading into the control address register an address from control memory, transferring an external address, or loading an initial address to start the control operations. The control data register holds the present microinstruction while the next address is computed and read from memory; the data register is sometimes called a pipeline register

6. Illustrate the mapping from micro-operation to a micro-instruction address.

A special type of branch exists when a microinstruction specifies a branch to the first word in control memory where a microprogram routine for instruction is located. The status bits for this type of branch are the bits in the operation code part of the instruction.

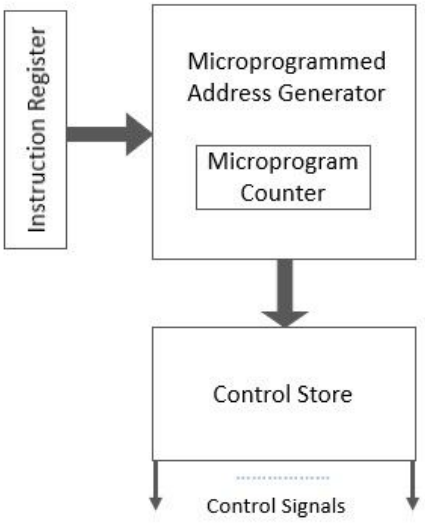
For example, a computer with a simple instruction format has an operation code of four bits which can specify up to 16 distinct instructions. Assume further that the control memory has 128 words, requiring an address of seven bits. For each operation code, there exists a microprogram routine in control memory that executes the instruction.

The contents of the mapping ROM 41 give the bits for the control address register. In this way, the microprogram routine that executes the instruction can be placed in any desired location in control memory. The mapping concept provides flexibility for adding instruction for control memory as the need arises.

The mapping function is sometimes implemented using an integrated circuit called programmable logic device or PLD. A PLD is similar to ROM in concept except that it uses AND and OR gates with internal electronic fuses. The interconnection between inputs, AND gates, OR gates, and outputs can be programmed as in ROM. A mapping function that can be expressed in terms of Boolean expressions can be implemented conveniently.

7. Explain the microprogram control with a diagram and give an example.

The basic components of a microprogrammed control MODULE are the control memory and the circuits that select the next address. The address selection part is called a microprogram sequencer. A microprogram sequencer can be constructed with digital functions to suit a particular application. However, just as there are large ROM MODULEs available in integrated circuit packages, so are general-purpose sequencers suited for the construction of microprogram control MODULEs. To guarantee a wide range of acceptability, an integrated circuit sequencer must provide an internal organization that can be adapted to a wide range of applications.



Micro instruction may be read and executed. The next-address logic of the sequencer determines the specific address source to be loaded into the control address register. The choice of the address source is guided by the next- address information bits that the

sequencer receives from the present microinstruction. Commercial sequencers include within the MODULE an internal register stack used for the temporary storage of addresses during microprogram looping and subroutine calls. some sequencers provide an output register which can function as the address register for the control memory.

8. Classify logic micro-operations and explain one stage of the logic circuit along with functional table

Logic micro-operations specify binary operations for strings of bits stored in registers. These operations consider each bit of the register separately and treat them as binary variables. For example, the exclusive-OR micro-operation with the contents of two registers R1 and R2 is symbolized by the statement

P: R1 ← R1 ⊕ R2

The 16 Boolean functions of two variables x and y are expressed in the algebraic form in the first column of Table 2-6. The 16 logic micro-operations are derived from these functions by replacing variable x with the binary content of register A and variable y by the binary content of register B. It is important to realize that the Boolean functions listed in the first column of Table 2 -6 represent a relationship between two binary variables x and y.

Logic micro-operations are very useful for manipulating individual bits or a portion of a word stored in a register. They can be used to

change bit values, delete a group of bits, or insert new bit values into a register. The following examples show how the bits of one register (designated by A) are manipulated by logic micro-operations as a function of the bits of another register (designated by B). In a typical application, register A is a processor register and the bits of register B constitute a logic operand extracted from memory and placed in register B.

9. Explain the logic of a microprogram sequencer for a control memory

The next address generator is sometimes called a microprogram sequencer, as it determines the address sequence that is read from control memory. The address of the next microinstruction can be specified in several ways, depending on the sequencer inputs. Typical functions of a microprogram sequencer are incrementing the control address register by one, loading into the control address register an address from control memory, transferring an external address, or loading an initial address to start the control operations.

The other two components: the sequencer and the control memory are combinational circuits and do not need a clock.

Micro instructions are stored in control memory in groups, with each group specifying routine. Each computer instruction has its own microprogram routine in control memory to generate the micro-operations that execute the instruction. The hardware that controls the address sequencing of the control memory must be

capable of sequencing the microinstructions within a routine and be able to branch from one routine to another

The next step is to generate the micro-operations that execute the instruction fetched from memory. The micro-operation steps to be generated in the processor register depend on the operation code part of the instruction. Each instruction has its own microprogram routine stored in a given location of control memory. The transformation from the instruction code bits to an address in control memory where the routine is located is referred to as a mapping process. A mapping procedure is a rule that transforms the instruction code into a control memory address. Once the required routine is reached, the microinstructions that execute the instruction may be sequenced by incrementing the control address register, but sometimes the sequence of micro operations will depend on values of certain status bits in processor.

10. Explain the rules in arithmetic operations on floating point numbers.

- Floating-point numbers (format F) are represented as a sum of powers of two (as are integer numbers (format I)), whereas unpacked and packed numbers (formats N and P) are represented as a sum of powers of ten.
- In unpacked or packed numbers, the position of the decimal point is fixed. In floating-point numbers, however, the position

of the decimal point (as the name indicates) is "floating", that is, its position is not fixed, but depends on the actual value.

- Floating-point numbers are essential for the computing of trigonometric functions or mathematical functions such as sinus or logarithm.

PART B

1. Illustrate register transfer language (RTL) with examples.

A digital system is an interconnection of digital hardware modules that accomplish a specific information-processing task. Digital systems vary in size and complexity from a few integrated circuits to a complex of interconnected and interacting digital computers.The modules are constructed from such digital components as registers, decoders, arithmetic elements, and control logic. The operations executed on data stored in registers are called micro operations. A micro operation is an elementary operation performed on the information stored in one or more registers. The symbolic notation used to describe the micro operation transfers among registers is called a Register Transfer Language (RTL). The term register transfer implies the availability of hardware logic circuits that can perform a stated micro operation and transfer the result of the operation to the same or another register. The word language is borrowed from programmers, who apply this term to programming languages. A programming language is a procedure for writing symbols to specify a given

computational process. A register transfer language is a system for expressing in symbolic form the micro operation sequences among the registers of a digital module. It is a convenient tool for describing the internal organization of digital computers in a concise and precise manner. It can also be used to facilitate the design process of digital systems.
Example: R2 ← R1
The content of R1 is copied into R2 without affecting the content of R1. It is an unconditional type of transfer operation.

2. Summarize the common bus system using multiplexers with a neat design

A typical digital computer has many registers, and paths must be provided to transfer information from one register to another. The number of wires will be excessive if separate lines are used between each register and all other registers in the system. A more efficient scheme for transferring information between registers in a multiple-register configuration is a common bus system. A bus structure consists of a set of common lines, one for each bit of a register, through which binary information is transferred one at a time. Control signals 23 determine which register is selected by the onus during each particular register transfer. One way of constructing a common bus system is with multiplexers. The multiplexers select the source register whose binary information is then placed on the bus. The construction of a bus system for four registers is shown in Figure. 2-3. Each register has four bits,

numbered 0 through 3. The bus consists of four 4 × 1 multiplexers each having four data inputs, 0 through 3, and two selection inputs, S1 and S0. In order not to complicate the diagram with 16 lines crossing each other, we use labels to show the connections from the outputs of the registers to the inputs of the multiplexers. For example, output 1 of register A is connected to input 0 of MUX 1 because this input is labelled A1. The diagram shows that the bits in the same significant position in each register are connected to the data inputs of one multiplexer to form one line of the bus. Thus MUX 0 multiplexes the four 0 bits of the registers, MUX 1 multiplexes the four 1 bits of the registers, and similarly for the other two bits. The two selection lines S1 and S0 are connected to the selection inputs of all four multiplexers. The selection lines choose the four bits of one register and transfer them into the four-line common bus. When S1 S0 = 00, the 0 data inputs of all four multiplexers are selected and applied to the outputs that form the bus. This causes the bus lines to receive the content of register A since the outputs of this register are connected to the 0 data inputs of the multiplexers. Similarly, register B is selected if S1 S0 = 01, and so on. Table shows the register that is selected by the bus for each of the four possible binary values of the selection lines. In general, a bus system will multiply k registers of n bits each to produce an n-line common bus. The number of multiplexers needed to construct the bus is equal to n, the number of bits in each register. The size of each multiplexer must be k × 1 since it multiplexes k data lines. For example, a common bus for eight registers of 16 bits each requires 16 multiplexers, one for each line in the bus. Each multiplexer must

have eight data input lines and three selection lines to multiplex one significant bit in the eight registers.

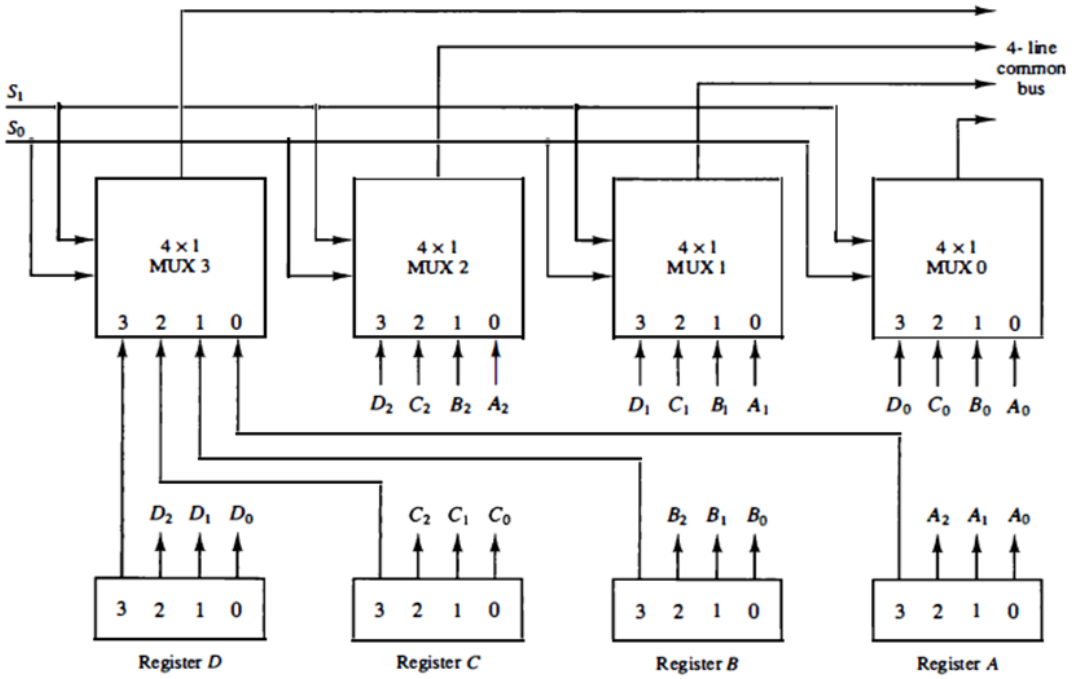


TABLE: Function Table for Bus of Fig

S_1	S_0	Register selected
0	0	*A*
0	1	*B*
1	0	*C*
1	1	*D*

3. Define micro-operation and illustrate arithmetic microoperations with examples.

The operations executed on data stored in registers are called micro-operations. A micro-operation is an elementary operation performed on the information stored in one or more registers.

Example: Shift, count, clear and load.

Types of Micro-Operations:

Register transfer micro-operations transfer binary information from one register to another.

Arithmetic micro-operations perform arithmetic operations on numeric data stored in registers.

Logic micro-operations perform bit manipulation operations on non-numeric data stored in registers.

Shift micro-operations perform shift micro-operations performed on data.

The basic arithmetic micro operations are addition, subtraction, increment, decrement, and shift. Arithmetic shifts are explained later in conjunction with the shift micro operations. The arithmetic micro operation defined by the statement specifies an add micro operation. It states that the contents of register R1 are added to the contents of register R2 and the sum transferred to register R3. R3 ← R1 + R2 To implement this statement with hardware we need three registers and the digital component that performs the addition operation. The other basic arithmetic micro operations are listed in Table 4-3. Subtraction is most often implemented through complementation and addition. Instead of using the minus operator, we can specify the subtraction by the following statement: R2 is the symbol for the 1's complement of R2. Adding 1 to the 1's complement produces the 2's complement. Adding the contents of R1 to the 2's complement of R2 is equivalent to R1 – R2

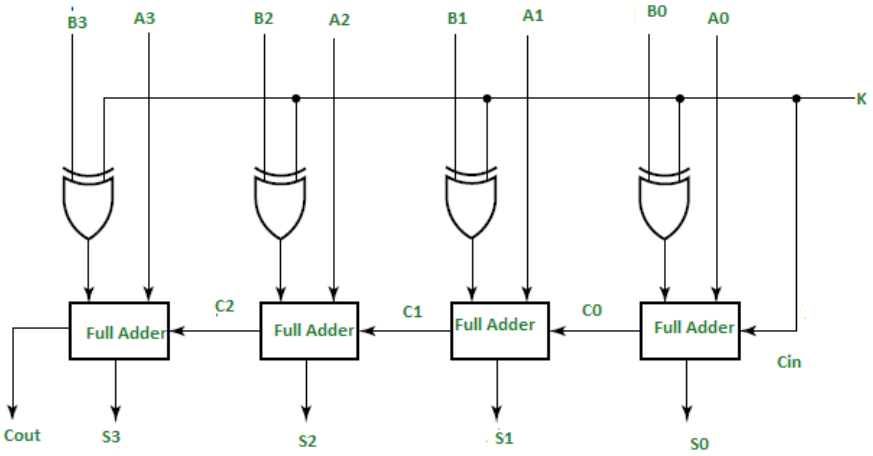
$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow \overline{R2}$	Complement the contents of R2
$R2 \leftarrow \overline{R2} + 1$	2's complement the contents of R2 (negate)
$R3 \leftarrow R1 + \overline{R2} + 1$	subtraction
$R1 \leftarrow R1 + 1$	Increment
$R1 \leftarrow R1 - 1$	Decrement

The increment and decrement micro operations are symbolized by plus-one and minus one operations, respectively. These micro operations are implemented with a combinational circuit or with a binary up-down counter.

4. Illustrate 4-bit binary adder subtractor along with a neat sketch.

The subtraction of binary numbers can be done most conveniently by means of complements. Remember that the subtraction A – B can be done by taking the 2's complement of B and adding it to A. The 2's complement can be obtained by taking the 1's complement and adding one to the least significant pair of bits. The 1's complement can be implemented with inverters and a one can be added to the sum through the input carry. The addition and subtraction operations can be combined into one common circuit by including an exclusive-OR gate with each full-adder. A 4 -bit adder-subtractor circuit is shown in Figure. The mode input M controls the operation. When M = 0 the circuit is an adder and when M = 1 the circuit becomes a subtractor. Each exclusive-OR gate

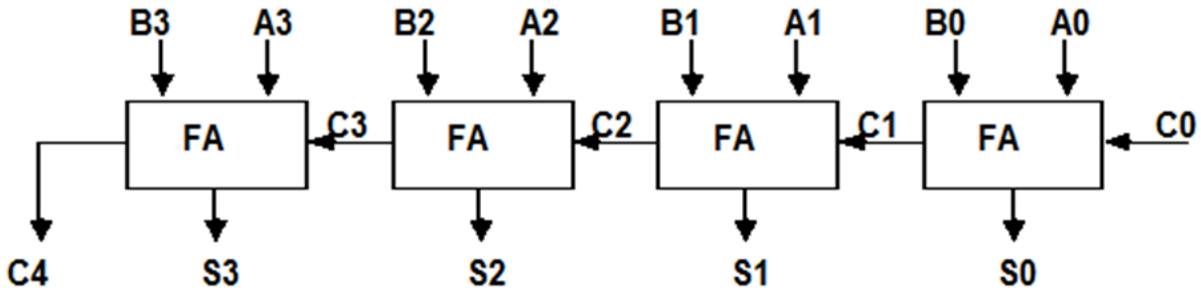
receives input M and one of the inputs of B. When M = 0, we have B ⊕ 0 = B. the full-adders receive the value of B, the input carry is 0, and the circuit performs A plus B. When M = 1, we have B ⊕ 1 = B' and C0 = 1. The B inputs are all complemented and a 1 is added through the input carry. The circuit performs the operation A plus the 2's complement of B. For unsigned, this gives A – B if A ≥ B or the 2's complement of (B−A) if A < B. For signed numbers, the result is A – B provided that there is no overflow.



5. Explain binary adder and 4-bit binary adder along with a neat sketch.

- Basic hardware required to implement add operation is registers and digital components that perform add operation.
- The digital circuit that forms the arithmetic sum of two bits and previous carry is called a full adder.
- The digital circuit that generates the arithmetic sum of two binary numbers of any length is called a binary adder.

- The binary adder is constructed with full adder circuits connected in cascade , with the output carry from one full adder connected to the input carry of the next full adder.
- An n bit binary adder requires n full adders.
- The output carry from each full adder is connected to the input carry of the next higher – order full adder.



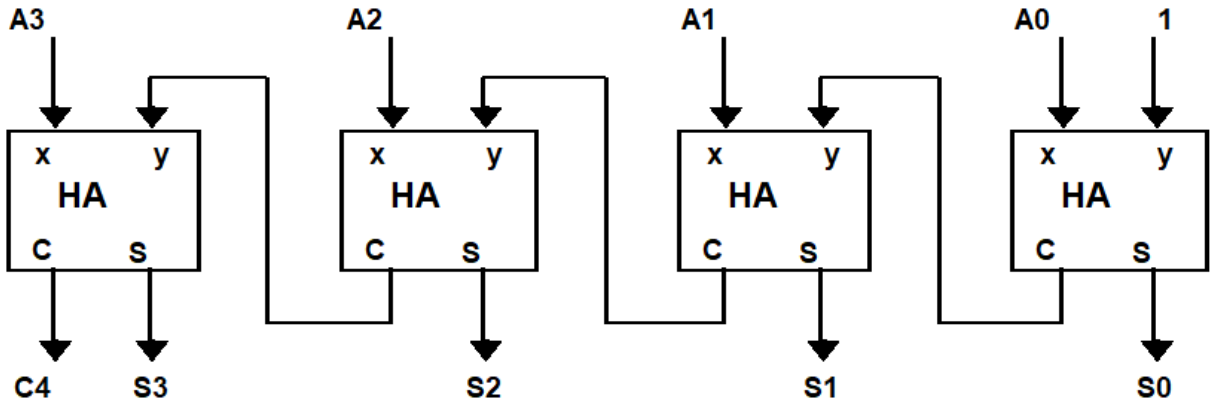
6. Explain 4-bit binary incrementer and along with a neat sketch.

The increment micro-operation adds one to a number in a register. For example, if a 4- bit register has a binary value 0110, it will go to 0111 after it is incremented. This microoperation is easily implemented with a binary counter. Every time the count enable is active, the clock pulse transition increments the content of the register by one. There may be occasions when the increment micro-operation must be done with a combinational circuit independent of a particular register. This can be accomplished by means of half-adders connected in cascade.

The diagram of a 4-bit combinational circuit incremented is shown in Figure. One of the inputs to the least significant half-adder (HA) is connected to logic-1 and the other input is connected to the least significant bit of the number to be incremented. The output carried

from one half-adder is connected to one of the inputs of the next-higher-order half adder.

The circuit receives the four bits from A0 through A3, adds one to it, and generates the incremented output in S0 through S3. The output carry C4 will be 1 only after incrementing binary 1111. This also causes outputs S0 through S3 to go to 0. The circuit of Figure can be extended to an N-bit binary incremented by extending the diagram to include n half-adders. The least significant bit must have one input connected to logic-. The other inputs receive the number to be incremented or the carry from the previous stage.



7. Explain multiple bus organizations in detail.

A bus is an important memory transferring device used by most computer devices and Smartphones to pass data back and forth across the entire system. The overall speed of the machine is directly affected by the types of buses used by it. Simple computer design uses single bus structures for transferring data, and Multiple bus organization uses multiple buses for enhanced performance.

In a multi-bus architecture, all the pathways are suited for handling some special types of information.

In single bus architecture, all the devices use a common bus for data transfer; thus, the system's efficiency and performance is lower. However, in the multiple bus organization, the wasted time lowers down, and thus the speed and performance of the entire system boost up. Thus, this is one of the key reasons behind using Multiple bus organizations.

Benefits of Multi Bus Organisation:

- Allows more number of devices to be connected to the computer
- Faster execution because many devices can work simultaneously resulting in performance enhancement
- Compatible with both old and new devices alike
- Multi-core processor that transfers more information and minimizes the wait time

8. Classify shift Micro operations and explain 4-bit combinational circuit shifters.

- Shift micro operations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic, and other data-processing operations.
- The information transferred through the serial input determines the type of shift. There are three types of shifts: logical, circular, and arithmetic.

4 bit combinational circuit shifter

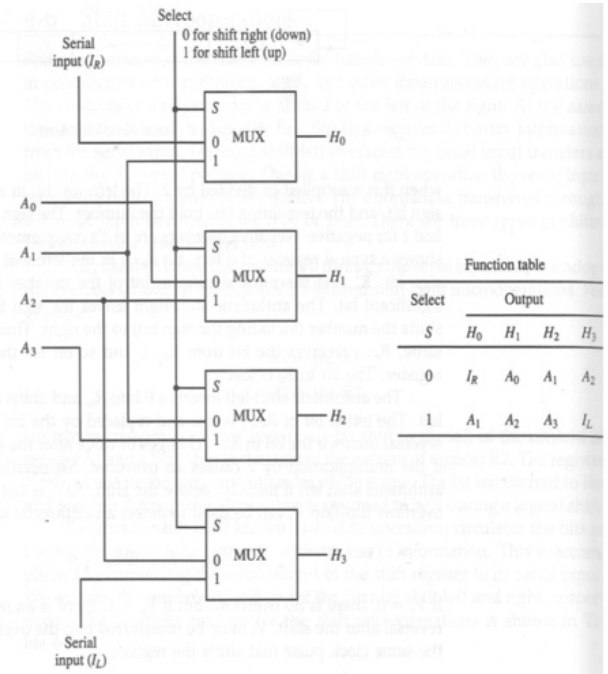


Figure 4-12 4-bit combinational circuit shifter.

9. Make use of control memory and build microprogrammed control organization

The general configuration of a microprogrammed control MODULE is demonstrated in the block diagram of Figure. The control memory is assumed to be a ROM, within which all control information is permanently stored. The control memory address register specifies the address of the microinstruction, and the control data register holds the microinstruction read from memory. The microinstruction contains a control word that specifies one or more micro operations for the data processor.

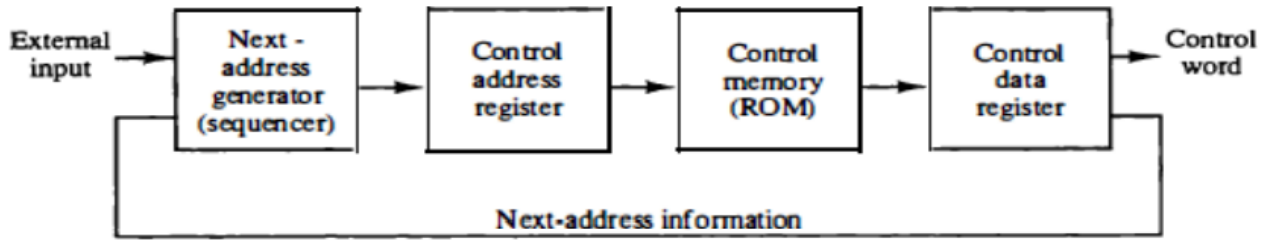


Fig 2.11. Micro programmed control organization

Once these operations are executed, the control must determine the next address. The location of the next microinstruction may be the one next in sequence, or it may be located somewhere else in the control memory. For this reason it is necessary to use some bits of the present microinstruction to control the generation of the address of the next microinstruction. The next address may also be a function of external input conditions. While the micro operations are being executed, the next address is computer in the next address generator circuit and then transferred into the control address register to read the next microinstruction. Thus a microinstruction contains bits for initiating micro operations in the data processor part and bits that determine the address sequence for the control memory.

The control data register holds the present microinstruction while the next address is computed and read from memory; the data register is sometimes called a pipeline register. It allows the execution of the micro operations specified by the control word simultaneously with the generation of the next microinstruction.The control word and next-address information are taken directly from the control memory. ;it must be realized that a ROM operates as a

combinational circuit, with the address value as the input and the corresponding word as the output.

The main advantage of the microprogrammed control is the fact that once the hardware configuration is established; there should be no need for further hardware or wiring changes. If we want to establish a different control sequence for the system, all we need to do is specify a different set of microinstructions for control memory.

10. What are the two approaches used for generating the control signals in proper sequence?

To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence. There are two approaches used for generating the control signals in proper sequence as Hardwired Control unit and Micro-programmed control unit.

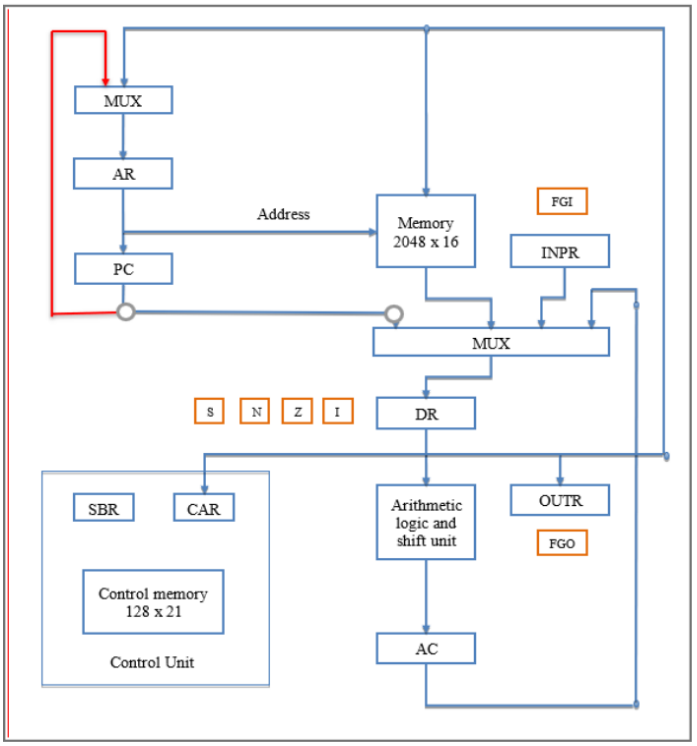
When the control signals are generated by hardware using conventional logic design techniques, the control MODULE is said to be hardwired. Microprogramming is a second alternative for designing the control MODULE of a digital computer. The principle of microprogramming is an elegant and systematic method for controlling the micro operation sequence in a digital computer.

11. Explain the mapping of instruction and subroutine in address sequencing.

A mapping procedure is a rule that transforms the instruction code into a control memory address. Once the required routine is reached, the microinstructions that execute the instruction may be sequenced by incrementing the control address register, but sometimes the sequence of micro operations will depend on values of certain status bits in processor registers. microprograms that employ subroutines will require an external register for storing the return address. Return addresses cannot be stored in ROM because the MODULE has no writing capability. When the execution of the instruction is completed, control must return to the fetch routine. This is accomplished by executing an unconditional branch microinstruction to the first address of the fetch routine. In summary, the address sequencing capabilities required in control memory are:

1. Incrementing of the control addresses register.
2. Unconditional branch or conditional branch, depending on status bit conditions.
3. A mapping process from the bits of the instruction to an address for control memory.
4. A facility for subroutine call and return.

12. Identify microprogram example and build a computer hardware configuration



The block diagram of the computer is shown in Figure. It consists of two memory MODULEs: a main memory for storing instructions and data, and a control memory for storing the microprogram. Four registers are associated with the processor MODULE and two with the control MODULE. The processor registers are program counter PC, address register AR, DR, and accumulator register AC.The function of these registers is similar to the basic computer. The control MODULE has a control address register CAR and a subroutine register SBR. The control memory and its registers are organized as a microprogrammed control MODULE, as shown in Figure. The transfer of information among the register in the processor is done through multiplexers rather than a common bus. DR can receive information from AC, PC, or memory. AR can receive information from PC or DR, PC can receive information only from AR. The arithmetic, logic, and shift MODULE performs micro operations

with data from AC and DR and places the result in AC. Note that memory receives its address from AR. 43 Input data written to memory come from DR, and data read from memory can go only to DR. The computer instruction format is depicted in Figure. It consists of three fields: a 1-bit field for indirect addressing symbolized by I, a 4-bit operation code (opcode), and an 11-bitaddress field. Figure 4-5(b) lists four of the 16 possible memory-reference instructions. The ADD instruction adds the content of the operand found in the effective address to the content of AC. The BRANCH instruction causes a branch to the effective address if the operand in AC is negative. The program proceeds with the next consecutive instruction if AC is not negative. The AC is negative if its sign bit (the bit in the leftmost position of the register) is a 1. The STORE instruction transfers the content of AC into the memory word specified by the effective address. The EXCHANGE instruction swaps the data between AC and the memory word specified by the effective address.

13. Identify the microinstruction format and also model fetch routine in the microprogram example.

14. Explain memory locations and addresses.

Memory locations and addresses determine how the computer's memory is organized so that the user can efficiently store or retrieve information from the computer. The computer's memory is made of a silicon chip which has millions of storage cells, where

each storage cell is capable of storing a bit of information whose value is either 0 or 1. The memory of the computer is organized in such a way that the group of these n bits can be stored and retrieved easily by the computer in a single operation. The group of n bits is termed as word where n is termed as the word length. The word length of the computer has evolved from 8, 16, 24, 32 to 64 bits. General-purpose computers nowadays have 32 to 64 bits. The group of 8 bits is called a byte.

15. Summarize the different types of addressing modes

There are various types of Addressing Modes which are as follows –

Implied Mode – In this mode, the operands are specified implicitly in the definition of the instruction. For example, the instruction "complement accumulator" is an implied-mode instruction because the operand in the accumulator register is implied in the definition of the instruction. All register reference instructions that use an accumulator are implied-mode instructions.

Register Mode – In this mode, the operands are in registers that reside within the CPU. The specific register is selected from a register field in the instruction. A k-bit field can determine any one of the 2k registers.

Register Indirect Mode – In this mode, the instruction defines a register in the CPU whose contents provide the address of the operand in memory. In other words, the selected register includes the address of the operand rather than the operand itself.

Auto Increment or Auto Decrement Mode & minuend - This is similar to the register indirect mode except that the register is incremented or decremented after (or before) its value is used to access memory. When the address stored in the register defines a table of data in memory, it is necessary to increment or decrement the register after every access to the table. This can be obtained by using the increment or decrement instruction.

Direct Address Mode – In this mode, the effective address is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction. In a branch-type instruction, the address field specifies the actual branch address.

Indirect Address Mode – In this mode, the address field of the instruction gives the address where the effective address is stored in memory. Control fetches the instruction from memory and uses its address part to access memory again to read the effective address.

Indexed Addressing Mode – In this mode, the content of an index register is added to the address part of the instruction to obtain the

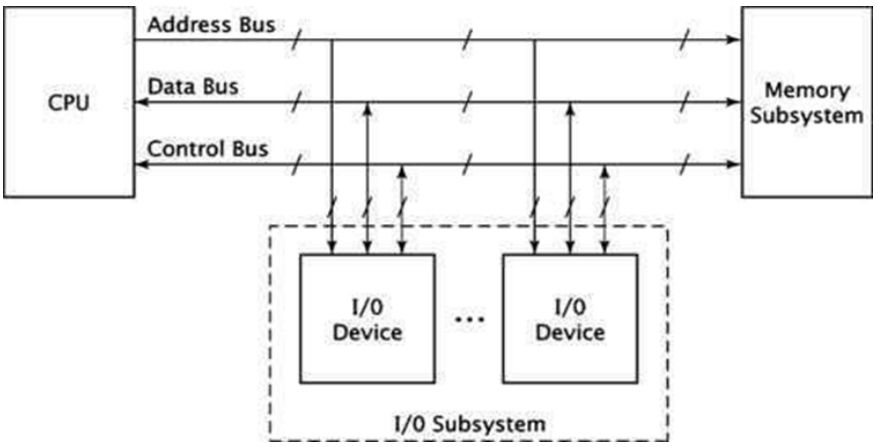
effective address. The index register is a special CPU register that contains an index value. The address field of the instruction defines the beginning address of a data array in memory.

16. What is a bus? Find the different bus with appropriate sketches.

The system bus has three buses:

- Address bus: The uppermost bus is the address bus. When the CPU reads data or instructions from or writes data to memory, it must specify the address of the memory location it wishes to access.

- Data bus: Data is transferred via the Data bus. When CPU fetches data from memory first outputs the memory address on to its address bus. Then memory outputs the data onto the data bus. Memory then reads and stores the data at the proper locations.



- Control bus: Control bus carries the control signal. Control signal is the collection of individual control signals. These signals indicate whether data is to be read into or written out of the CPU.

18. Explain about the microprogram control unit with a neat sketch.

Refer Q9 in Part B

19. Illustrate register transfer language (RTL) with examples.

Refer Q1 in Part B

20. Summarize the common bus system using multiplexers with a neat sketch.

Refer Q16 in Part B

PART C

1. Define the register transfer language

The symbolic notation used to describe the micro operation transfers among registers is called a register transfer language. The term register transfer implies the availability of hardware logic circuits that can perform a stated micro operation and transfer the result of the operation to the same or another register. The word language is borrowed from programmers, who apply this term to

programming languages. A programming language is a procedure for writing symbols to specify a given computational process.
A register transfer language is a system for expressing in symbolic form the micro operation sequences among the registers of a digital module. It is a convenient tool for describing the internal organization of digital computers in a concise and precise manner. It can also be used to facilitate the design process of digital systems

2. What is bus and memory transfer

A typical digital computer has many registers, and paths must be provided to transfer information from one register to another. A bus structure consists of a set of common lines, one for each bit of a register, through which binary information is transferred one at a time. Control signals 23 determine which register is selected by the onus during each particular register transfer.
Many computers use one common bus to transfer information between memory or I/O and the CPU. The distinction between a memory transfer and I/O transfer is made through separate read and write lines. The CPU specifies whether the address on the address lines is for a memory word or for an interface register by enabling one of two possible read or write lines. The I/O read and I/O writes control lines are enabled during an I/O transfer. The memory read and memory write control lines are enabled during a memory transfer

3. State the arithmetic microoperations.

Arithmetic micro operations perform arithmetic operations on numeric data stored in registers.

The basic arithmetic micro operations are addition, subtraction, increment, decrement, and shift. Arithmetic shifts are explained later in conjunction with the shift micro operations. The arithmetic micro operation defined by the statement specifies an add micro operation. It states that the contents of register R1 are added to the contents of register R2 and the sum transferred to register R3.

$R3 \leftarrow R1 + R2$

For more(pg no 27 LEC NOTES)

4. What is the need for a register?

Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.

A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).

The computer needs processor registers for manipulating data and a register for holding a memory address. The register holding the memory location is used to calculate the address of the next instruction after the execution of the current instruction is completed.

5. State the shift micro-operations

Shift micro-operations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic, and other data-processing operations. The contents of a register can be shifted to the left or the right. At the same time that the bits are shifted, the first flip-flop receives its binary information from the serial input. During a shift-left operation the serial input transfers a bit into the right most position. During a shift-right operation the serial input transfers a bit into the leftmost position. The information transferred through the serial input determines the type of shift. There are three types of shifts: logical, circular, and arithmetic. A logical shift is one that transfers 0 through the serial input. We will adopt the symbols shl and shr for logical shift- left and shift-right micro-operations. For example:

R1 ← Sh1 R1

R2 ←shr R2

6. What is meant by hardwired control?

When the control signals are generated by hardware using conventional logic design techniques, the control MODULE is said to be hardwired .It should be mentioned that must computers based on the reduced instruction set computer (RISC) architecture concept, use hardwired control rather than a control memory with a microprogram.

The main advantage of the microprogrammed control is the fact that once the hardware configuration is established; there should

be no need for further hardware or wiring changes. If we want to establish a different control sequence for the system, all we need to do is specify a different set of microinstructions for control memory.

7. Explain tri state buffer with their application

The data from the input device goes to the tri-state buffers. When the value in the address and control buses are correct, the buffers are enabled and data passes on the data bus Tri-state buffers are replaced by a register. The tri-state buffers are used in input device interfaces to make sure that one device writes data to the bus at any time.

8. Find control addresses registered in control memory.

While the micro operations are being executed, the next address is computer in the next address generator circuit and then transferred into the control address register to read the next microinstruction. Thus a microinstruction contains bits for initiating micro operations in the data processor part and bits that determine the address sequence for the control memory.

Typical functions of a microprogram sequencer are incrementing the control address register by one, loading into the control address register an address from control memory, transferring an external address, or loading an initial address to start the control operations.

A reference to this bit by the status bit select lines from control memory causes the branch address to loaded into the control address register unconditionally

9. How is the mapping process in address sequencing?the address sequencing capabilities required in control memory are:

A mapping process from the bits of the instruction to an address for control memory. One simple mapping process that converts the 4-bit operation code to a 7-bit address for control memory is shown in fig. 4-3. This mapping consists of placing a 0 in the most significant bit of the address, transferring the four operation code bits, and clearing the two least significant bits of the control address register. This provides for each computer instruction a microprogram routine with a capacity of four microinstructions. If the routine needs more than four microinstructions, it can use addresses 1000000 through 1111111. If it uses fewer than four microinstructions, the unused memory cautions would be available for other routines; one can extend this concept to a more general mapping rule by using a ROM to specify the mapping function.

10. Tell the branch logic in address sequencing

The branch logic provides decision-making capabilities in the control MODULE. The status conditions are special bits in the system that provide parameter information such as the carry-out of an adder, the sign bit of a number, the mode bits of an instruction,

and input or output status conditions. Information in these bits can be tested and actions initiated based on their condition: whether their value is 1 or 0. The status bits, together with the field in the microinstruction that specifies a branch address, control the conditional branch decisions generated in the branch logic. The branch logic hardware may be implemented in a variety of ways. The simplest way is to test the specified condition and branch to the indicated address if the condition is met; otherwise, the address register is incremented.

11. Classify the instruction format.

A. Instruction is of variable length depending upon the number of addresses it contains. Generally, CPU organization is of three types based on the number of address fields:

- Single Accumulator organization
- General register organization
- Stack organization

In the first organization, the operation is done involving a special register called the accumulator.
Second, multiple registers are used for computation purposes.
In the third organization the work on stack basis operation due to which it does not contain any address field. Only a single organization doesn't need to be applied, a blend of various organizations is mostly what we see generally.

12. Define micro-operations in microprogram examples.

The operations executed on data stored in registers are called micro-operations. A micro-operation is an elementary operation performed on the information stored in one or more registers.
Example: Shift, count, clear, and load

13. What are the types of microinstruction?

There are 3 types of micro instruction.

Horizontal Microcode:

Each control signal is represented by a single bit in the control word. Thus, if the design has 500 control signals, this will require 500 bits in each control word to store the control bits. In this format, the control store looks horizontal in shape since the control words are wide.

Vertical Microcode:

The vertical microcode organization provides slower implementation compared to the horizontal microcode organization. To generate the control signals, it requires reading the CW code from the control store, decoding the CW, and then using OR gates.

Filed-Encoded Format:

It can be observed that there are signals in the design that cannot be 1 at the same time. These signals are called Mutually Exclusive signals. In order to save the size of the control store, all mutually exclusive signals can be grouped together and encoded so that a code is stored for this group instead of the actual signals. Then, a

decoder can be used to decode the group code and generate the signals.

14.Define Decode.

A. Converting the coded message into an understandable language.

15. State what is meant by full adder.

A. A full adder is a digital circuit that performs addition. Full adders are implemented with logic gates in hardware. A full adder adds three one-bit binary numbers, two operands and a carry bit.

16. Define addressing modes.

A. Techniques to specify the data in the instruction is called Addressing nodes.

17. What is a Bus? Draw the single bus structure.

A distinct set of conductors carrying data and control signals within a computer system, to which pieces of equipment may be connected in parallel.

18. What are the basic operations performed by the processor?

A computer is a device that transforms data into meaningful information. It processes the input according to the set of instructions provided to it by the user and gives the desired output quickly. A Computer can perform the following set of functions:

- Accept data
- Store data
- Process data as desired
- Retrieve the stored data as and when required
- Print the result in the desired format.

19. Demonstrate the principle operation of a microprogrammed control unit.

The logic of the control unit is specified by a micro-program. A microprogram is also called firmware (midway between the hardware and the software). It consists of:

- One or more micro-operations to be executed, and
- The information about the micro-instruction to be executed next.

20. Define the register transfer language.

The Register Transfer Language is the symbolic representation of notations used to specify the sequence of micro-operations. In a computer system, data transfer takes place between processor registers and memory and between processor registers and input-output systems.

Most of the Answers are taken from Lecture Notes and Google. Due to very limited time, all the relevant content has been dumped into this file, without any editing whatsoever. Only REFER this document for understanding, and not write it as is in the examination.