Basic Processing Unit

Fundamental Concepts

Processor fetches one instruction at a time and perform the operation specified.

Instructions are fetched from successive memory locations until a branch or a jump instruction is encountered.

Processor keeps track of the address of the memory location containing the next instruction to be fetched using Program Counter (PC).

Instruction Register (IR)

Executing an Instruction

Fetch the contents of the memory location pointed to by the PC. The contents of this location are loaded into the IR (fetch phase).

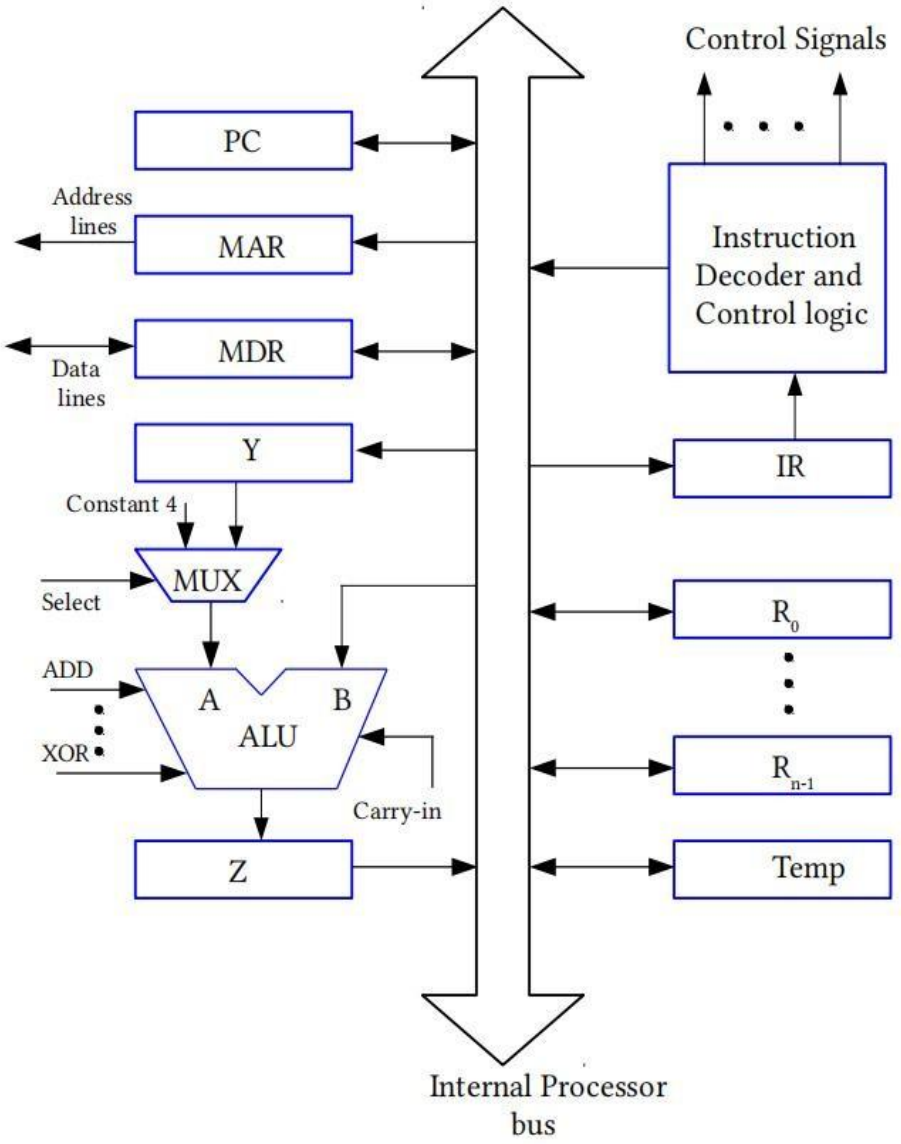
$$IR \leftarrow [[PC]]$$

Assuming that the memory is byte addressable, increment the contents of the PC by 4 (fetch phase).

$$PC \leftarrow [PC] + 4$$

Carry out the actions specified by the instruction in the IR (execution phase).

Single Bus CPU Organization



Basic Operations involved in the execution of an instruction

Transfer a word of data from one processor register to another or to the ALU.

Perform an arithmetic or a logic operation and store the result in a processor register.

Fetch the contents of a given memory location and load them into a processor register.

Store a word of data from a processor register into a given memory location.

Register Transfer Operation

MOV R1, R2

1. $R1_{out}$, $R2_{in}$

Fetching a Word from Memory

MOV (R1),R2

Step 1: Transfer the address into MAR;

Step 2: Issue Read operation;

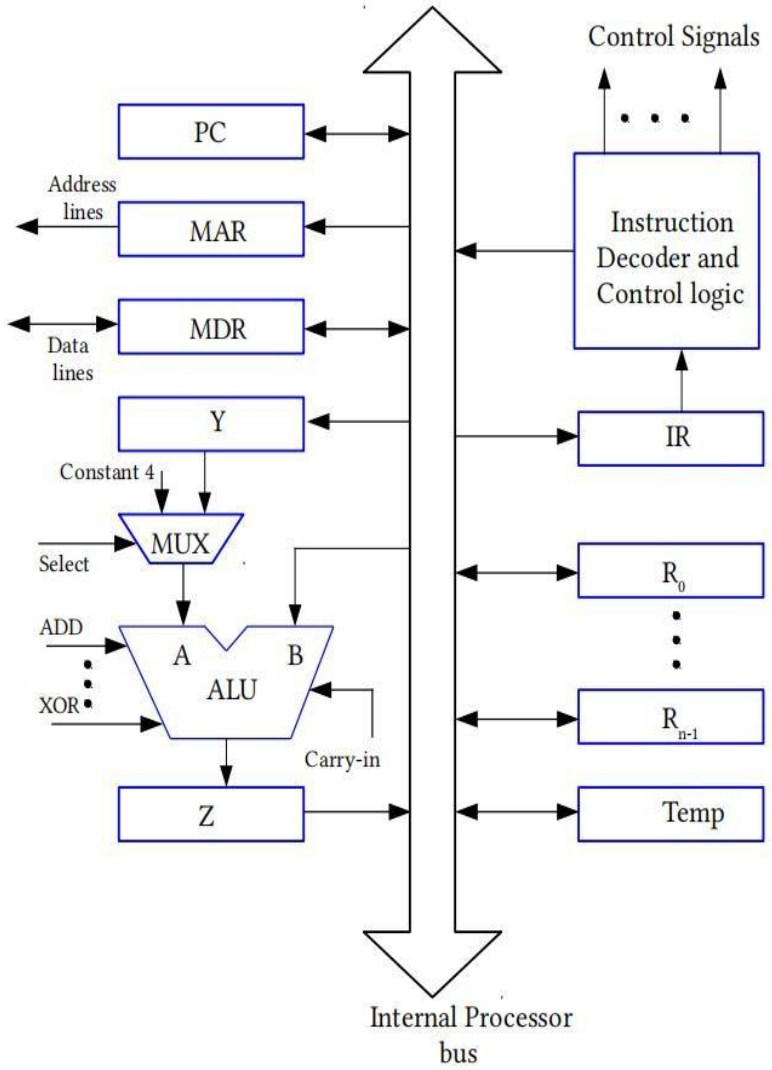
Step 3: Wait for data from memory

Step 4: Data from MDR will be transferred to the destination.

1. $R1_{out}$, MAR_{in}, Read

2. MDR_{inE}, WMFC

3. MDR_{out}, $R2_{in}$



Storing a Word into Memory

MOV R2, (R1)

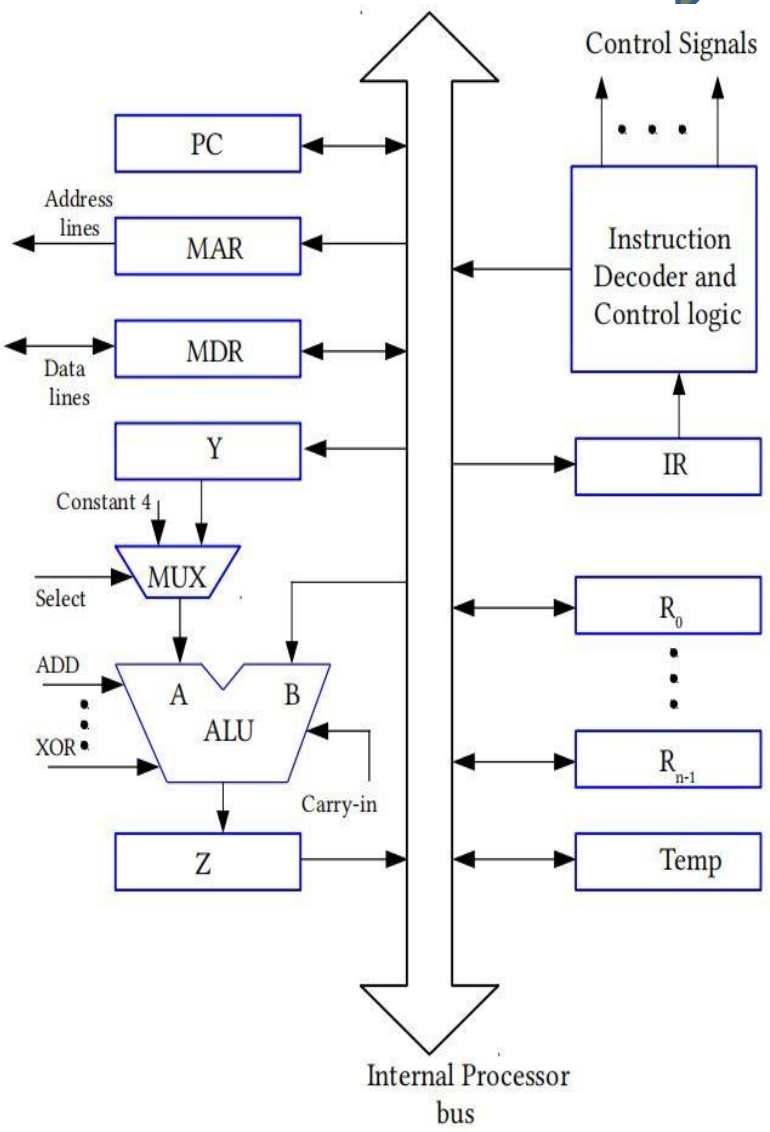
Address into MAR;

Data into MDR

Issue Write operation;

Wait for Write operation to complete

1. $R1_{out}$, MAR_{in}
2. $R2_{out}$, MDR_{in}, Write
3. MDR_{outE}, WMFC



Performing an Arithmetic or Logical Operation

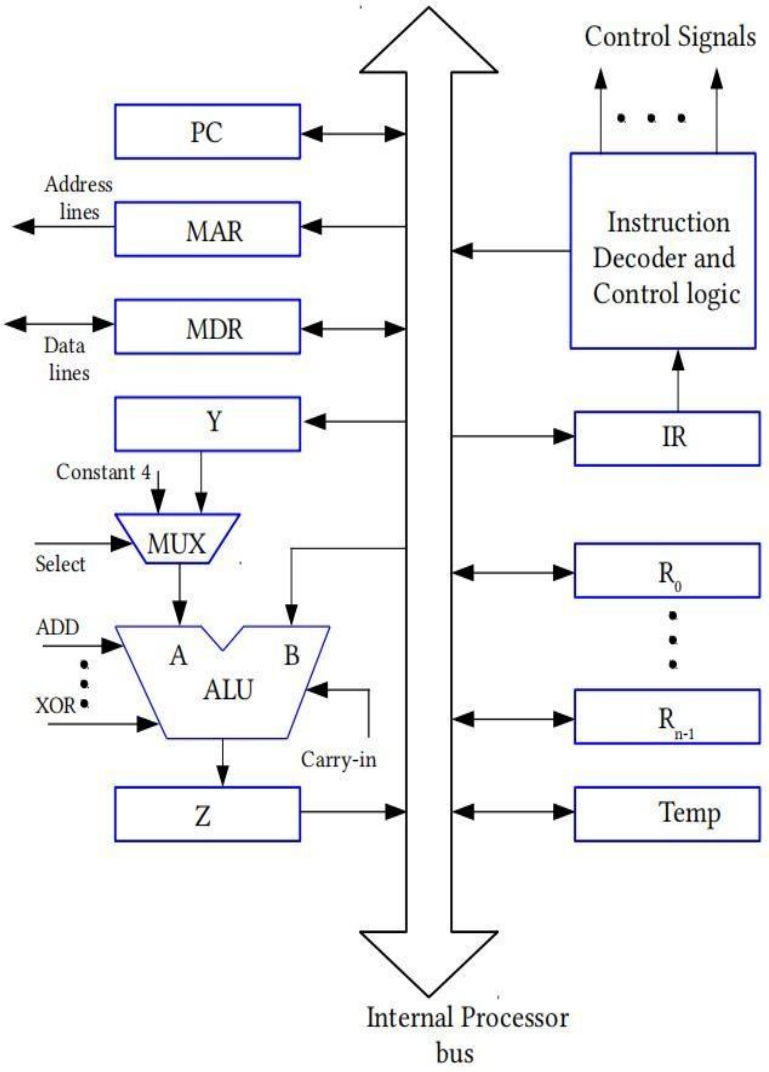
ADD R1, R2, R3

Put one of the operands into Y register

Put the other operand on the bus and perform the operation

Send the result into the destination

1. $R1_{out}$, Y_{in}
2. $R2_{out}$, SelectY, Add, Z_{in}
3. Z_{out}, $R3_{in}$



Execution of a Complete Instruction

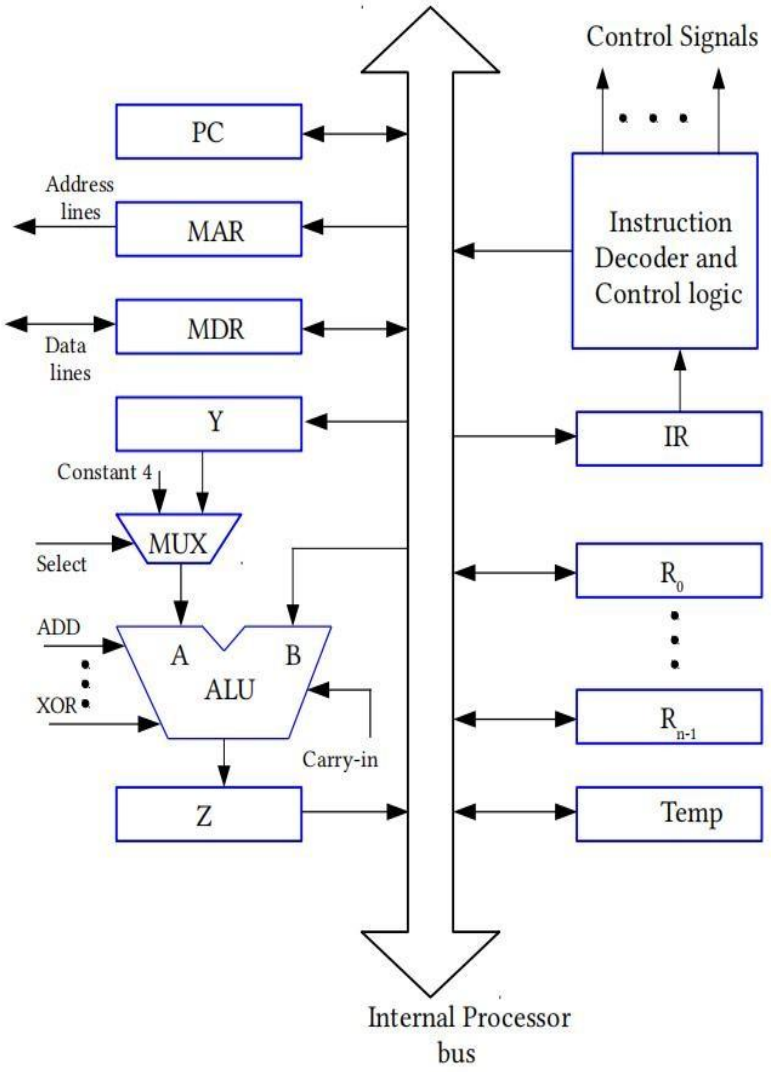
Add (R3), R1

Fetch the instruction

Fetch the first operand (the contents of the memory location pointed to by R3)

Perform the addition

Store the result into R1



Execution of a Complete Instruction

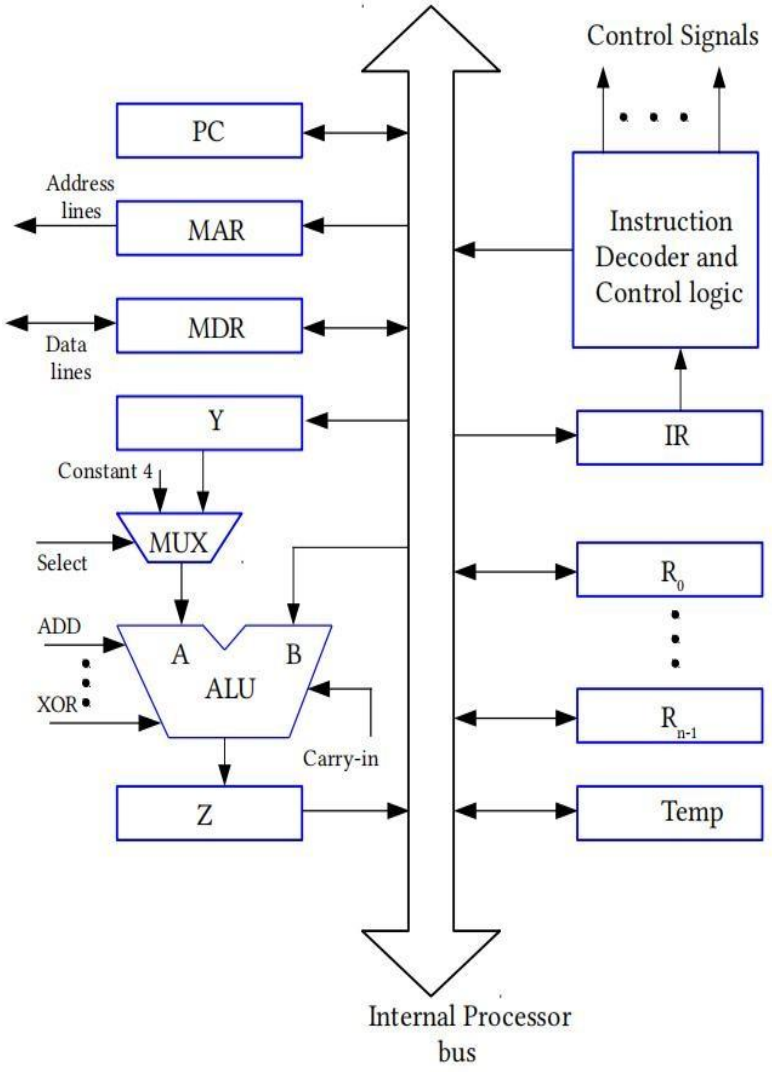
Add (R3), R1

Fetch the instruction

Fetch the first operand (the contents of the memory location pointed to by R3)

Perform the addition

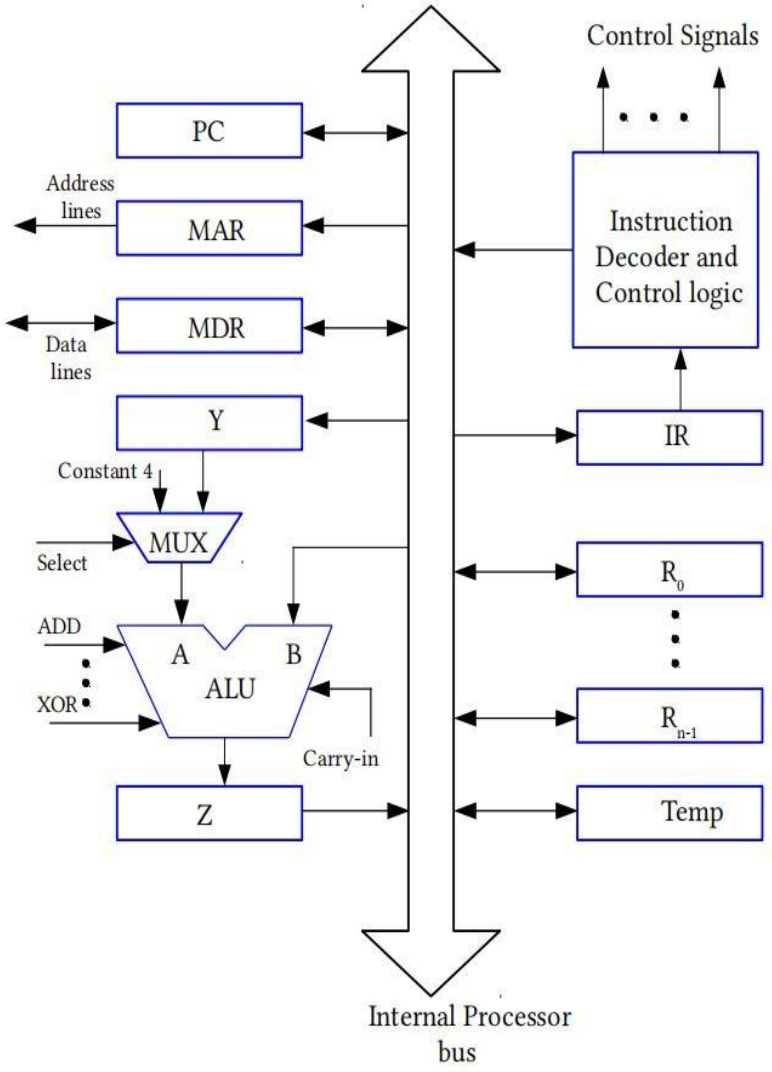
Store the result into R1



Execution of a Complete Instruction

Add (R3), R1

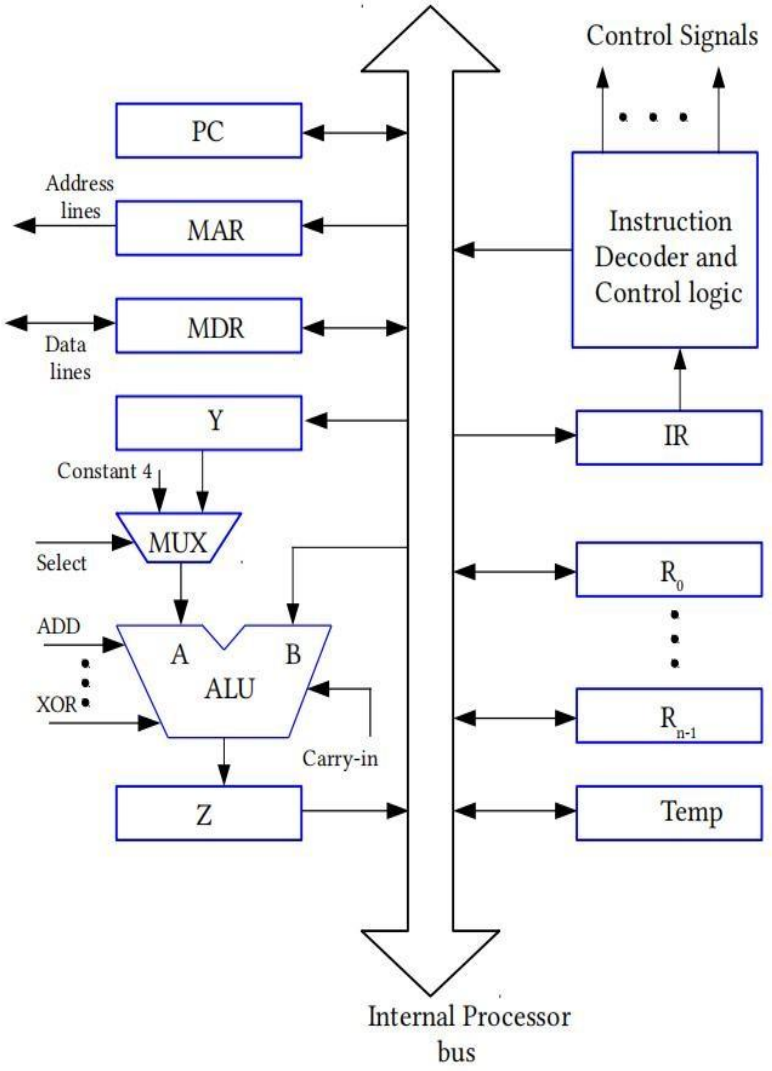
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, SelectY, Add, Zin
7. Zout, R1in, end



Execution of a Complete Instruction:Register Indirect Mode

Add R1, (R3)

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, SelectY, Add, Zin
7. R3out, MARin,
8. Zout, MDRin, Write
9. WMFC
10. End

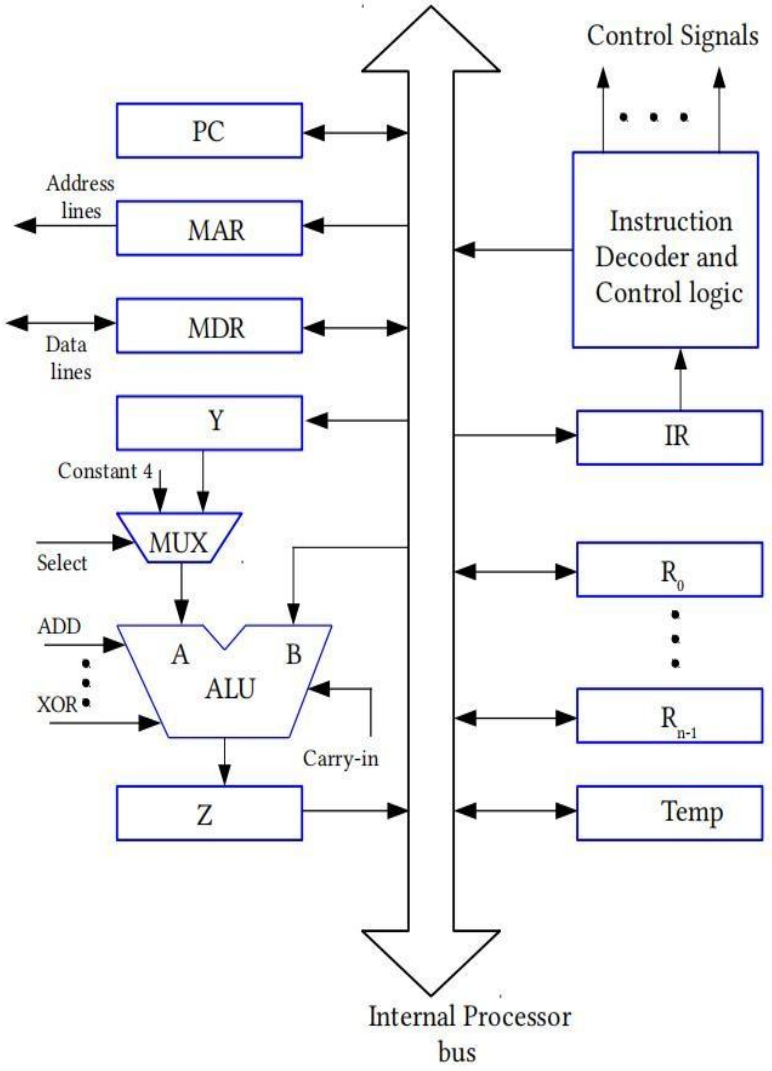


Problems(Immediate Mode)

Write the sequence of control steps required for the given bus structur for the following instruction:

Add the (immediate) number NUM to register R1

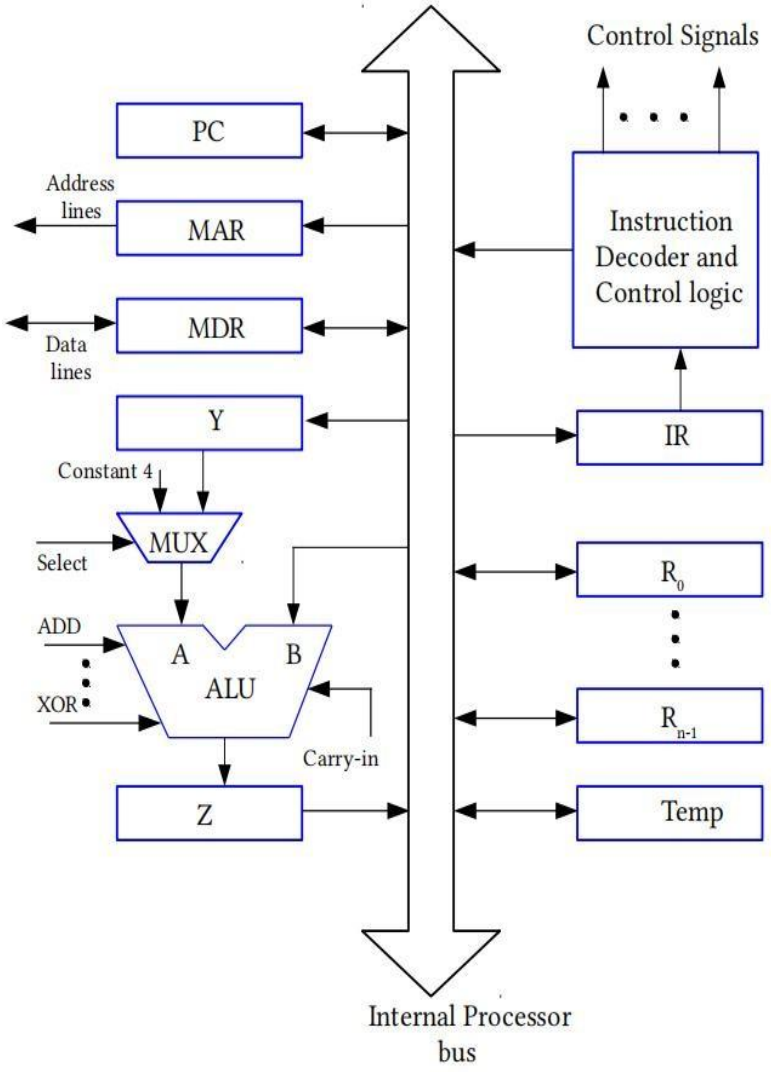
*Assume that each instruction consists of **two** words The **first** word specifies the operation and addressing mode, and the **second** word contains the number NUM*



Execution of a Complete Instruction : Immediate mode [2-word instruction]

ADD #NUM, R1

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. PCout, MARin, Read, Select 4, ADD, Zin
5. Zout, PCin, Yin,
6. R1out, Yin, WMFC
7. MDRout, SelectY, ADD, Zin
8. Zout, R1in, end

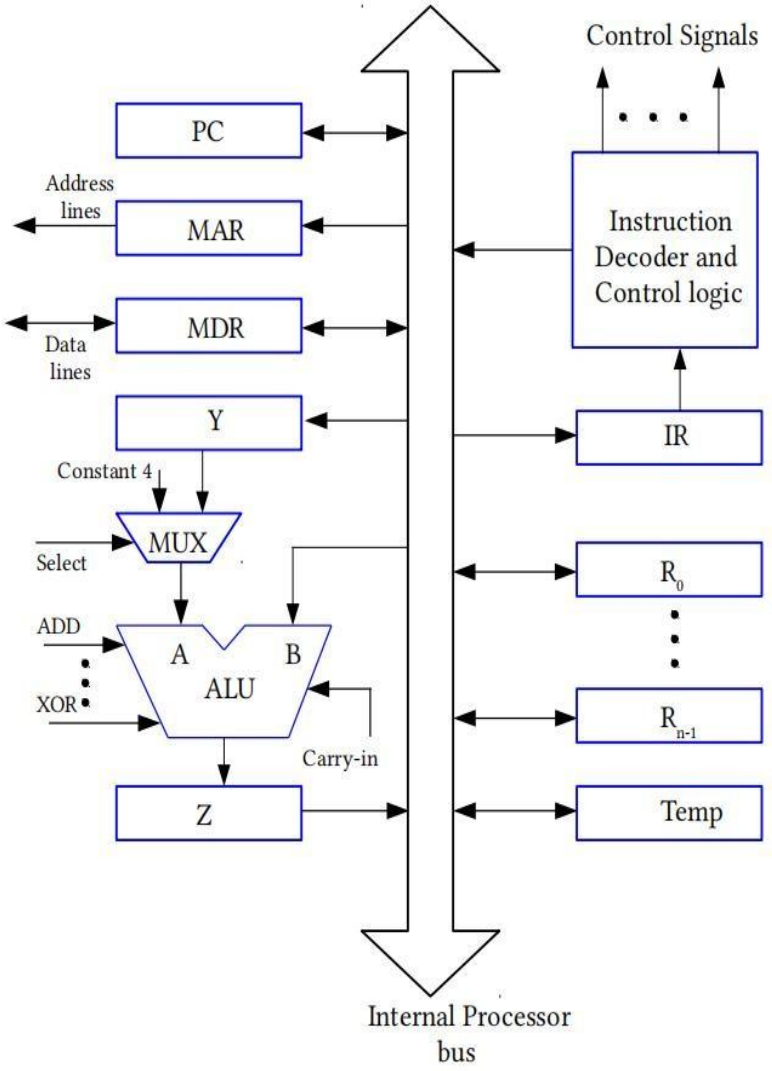


Problems (Absolute/Direct)

Write the sequence of control steps required for the given bus structur for the following instruction:

Add the contents of memory location NUM to register R1

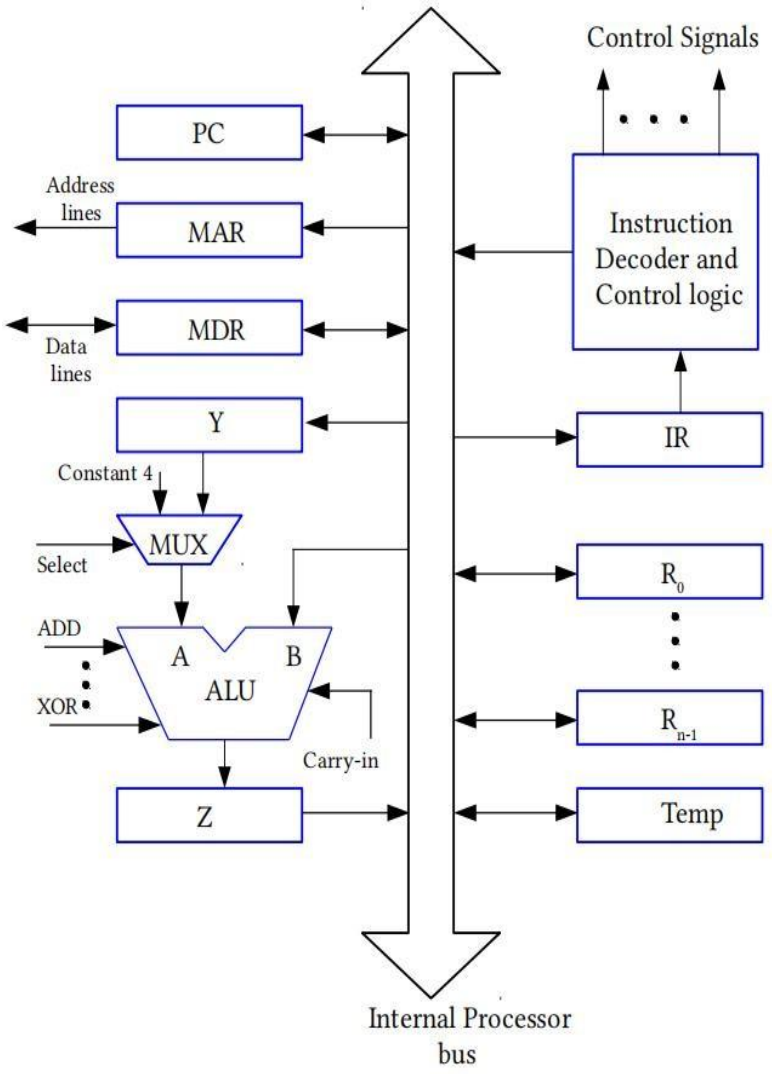
*Assume that each instruction consists of **two** words The **first** word specifies the operation and addressing mode, and the **second** word contains the number NUM*



Execution of a Complete Instruction : Direct/Absolute mode [2-word instruction]

ADD NUM, R1

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. PCout, MARin, Read, Select 4, ADD, Zin
5. Zout, PCin, Yin, WMFC
6. MDRout, MARin, Read
7. R1out, Yin, WMFC
8. MDRout, Select Y, Add, Zin
9. Zout, R1in, end

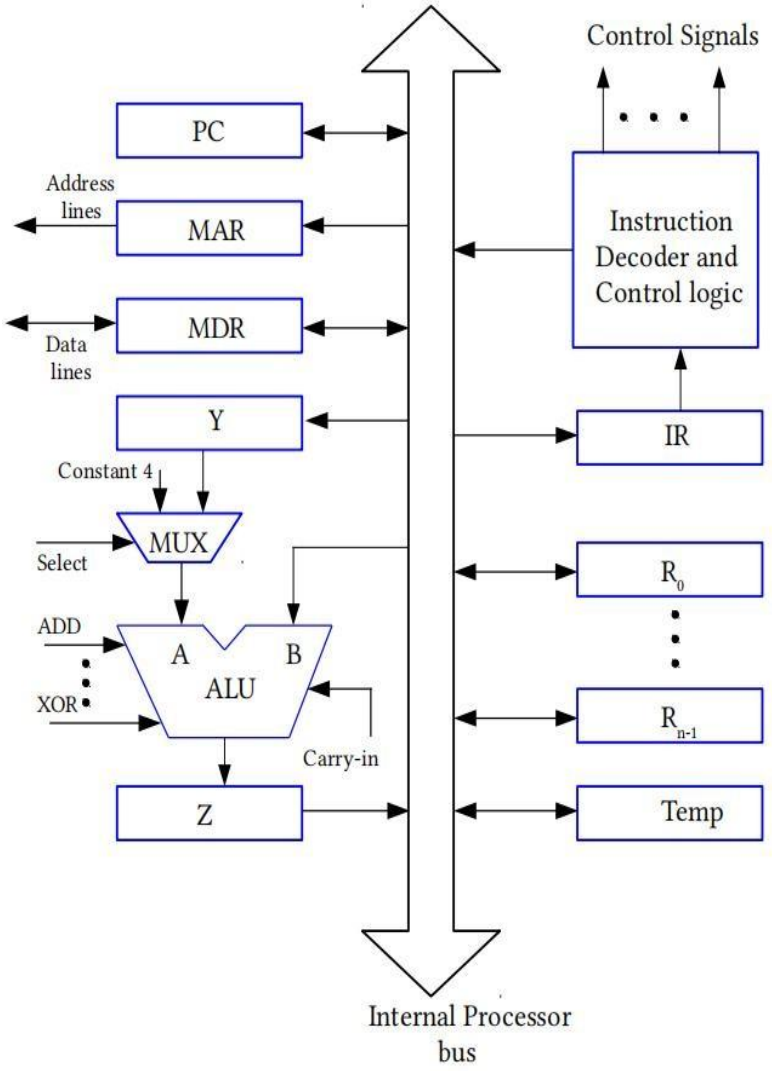


Problem: Memory Indirect Mode

Write the sequence of control steps required for the given bus structur for the following instruction:

Add the contents of memory location whose address is at memory location NUM to register R1

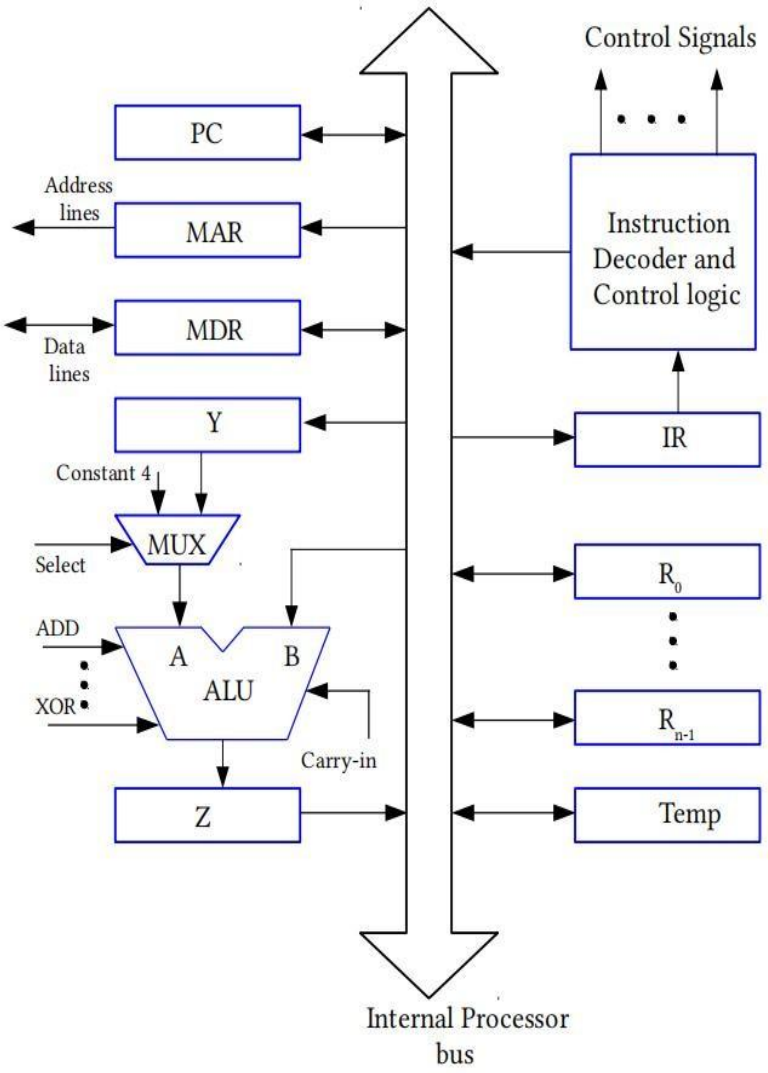
*Assume that each instruction consists of **two** words The **first** word specifies the operation and addressing mode, and the **second** word contains the number NUM*



Execution of a Complete Instruction : Memory Indirect mode [2-word instruction]

ADD (NUM), R1

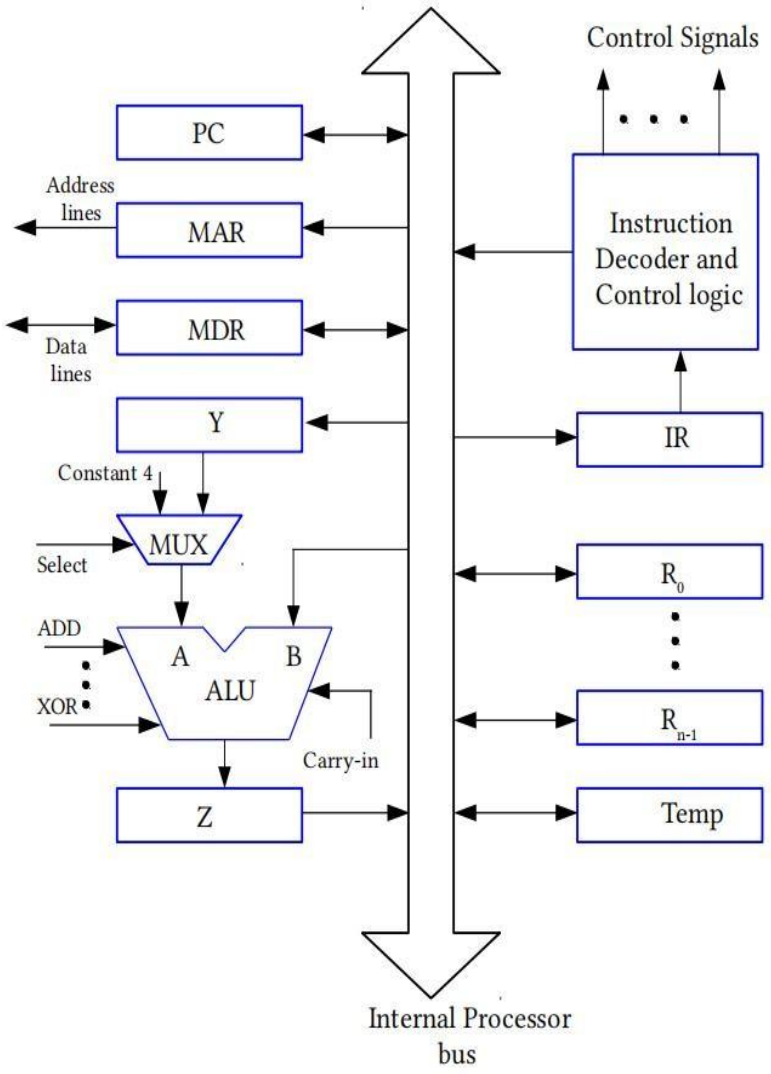
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. PCout, MARin, Read, Select 4, ADD, Zin
5. Zout, PCin, Yin, WMFC
6. MDRout, MARin, Read
7. WMFC
8. MDRout, MARin, Read,
9. R1out, Yin, WMFC
10. MDRout, SelectY, Add, Zin
11. Zout, R1in, end



Execution of a Complete Instruction : Direct mode [One word instruction]

MOV 300, R1

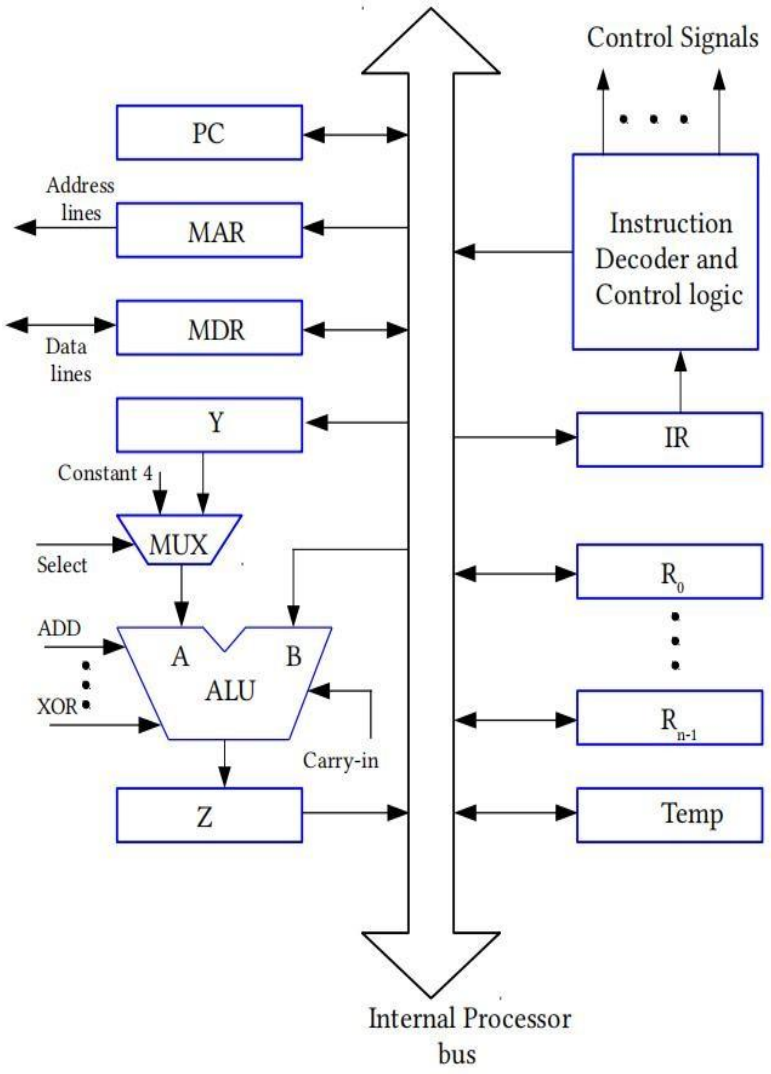
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address_Field_of_IRout, MARin, Read
5. WMFC
6. MDRout, R1in, end



Execution of a Complete Instruction : Indirect mode [One word instruction]

MOV (300), R1

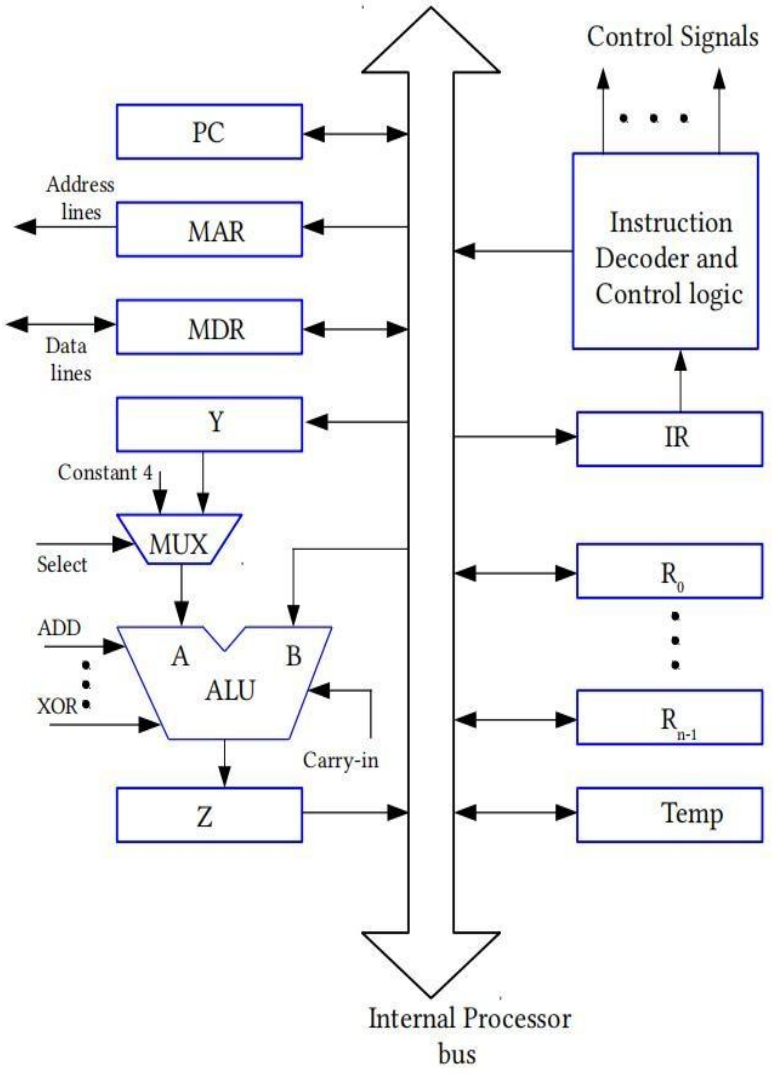
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address_Field_of_IRout, MARin, Read
5. WMFC
6. MDRout, MARin, Read
7. WMFC
8. MDRout, R1in, end



Execution of a Complete Instruction : Index mode [One word instruction]

Add 30(R1), R2

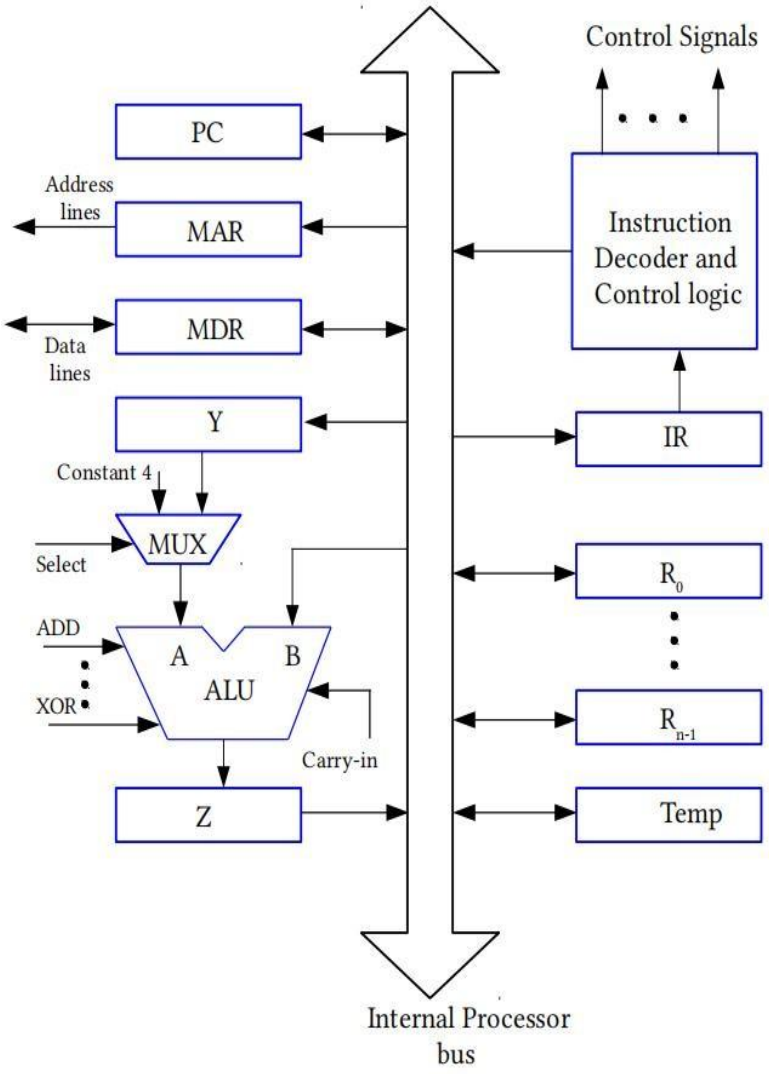
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R1out, Yin
5. Address_Field_of_IRout, SelectY, Add, Zin
6. Zout, MARin, Read
7. R2out, Yin, WMFC
8. MDRout, SelectY, Add, Zin
9. Zout, R2in, end



Execution of a Complete Instruction : Index mode [Two word instruction]

Add 30(R1), R2

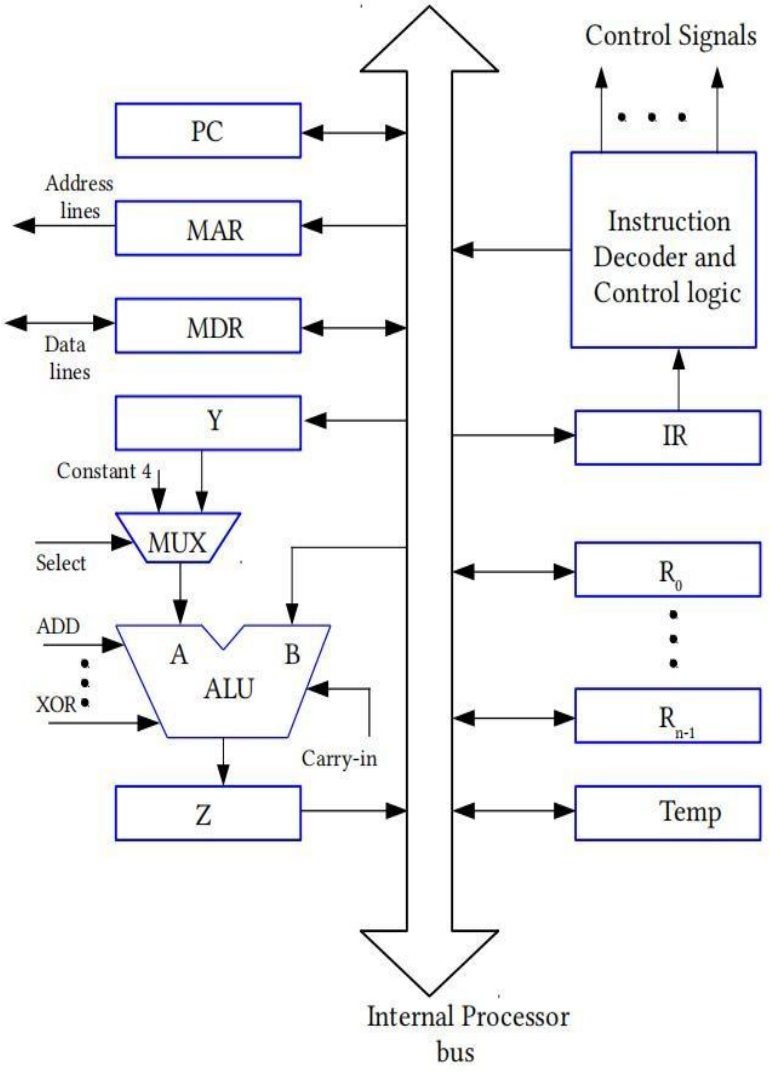
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. PCout, MARin, Read, Select4, Add, Zin
5. Zout, PCin, Yin
6. R1out, Yin, WMFC
7. MDRout, SelectY, Add, Zin
8. Zout, MARin, Read
9. R2out, Yin, WMFC
10. MDRout, SelectY, ADD, Zin



Execution of a Complete Instruction : Relative mode [One word instruction]

JMP L1

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address_Field_of_IRout, SelectY, Add, Zin
5. Zout, PCin, end

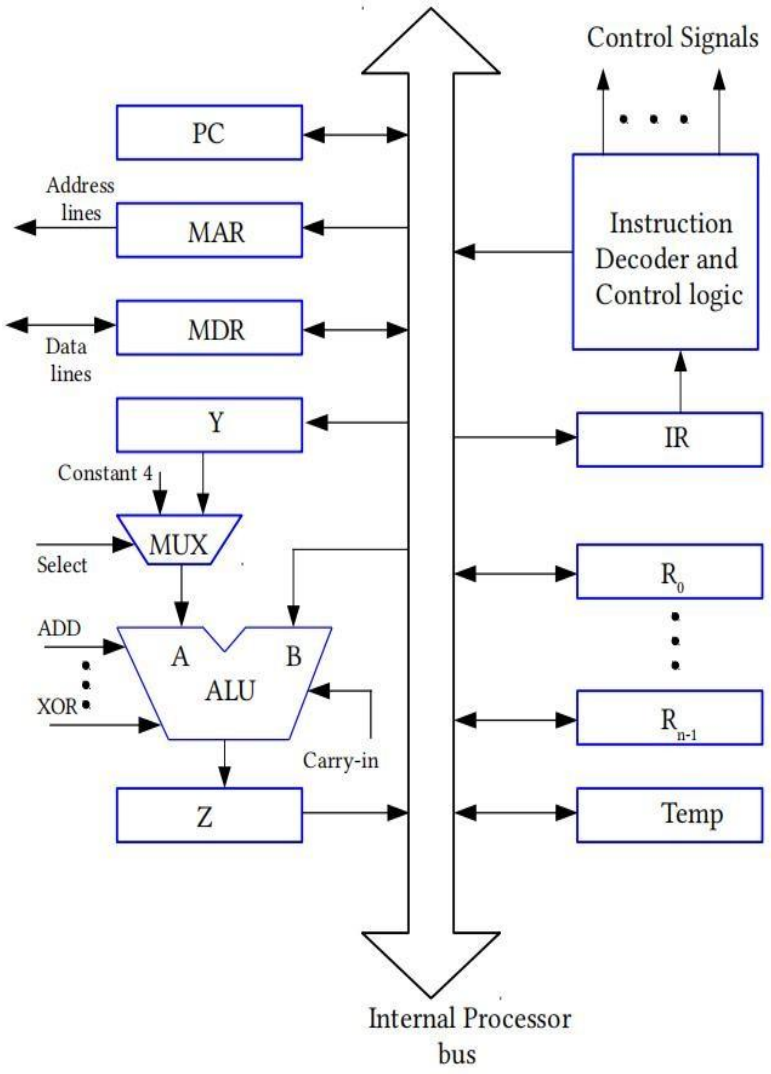


Execution of a Complete Instruction : Relative mode [One word instruction]

BR<0 L1

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address_Field_of_IRout, SelectY, Add, Zin,

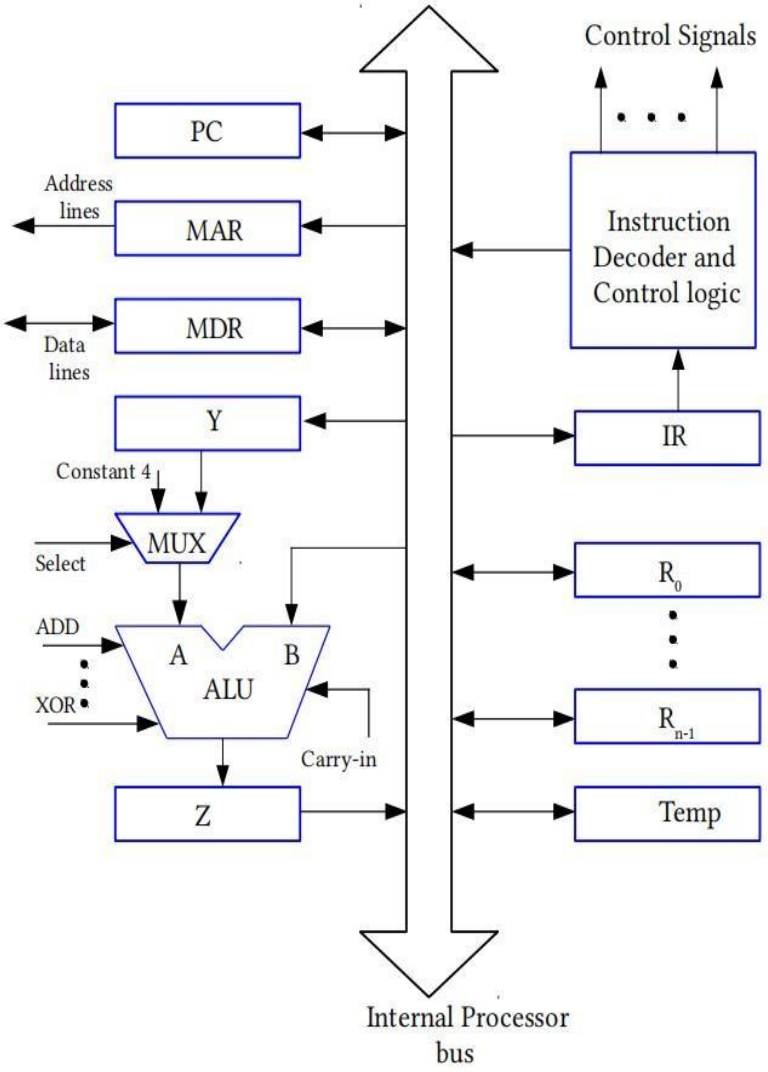
   N==0, then end
5. Zout, PCin, end



Execution of a Complete Instruction : Autodecrement mode

MUL -(R1), R2

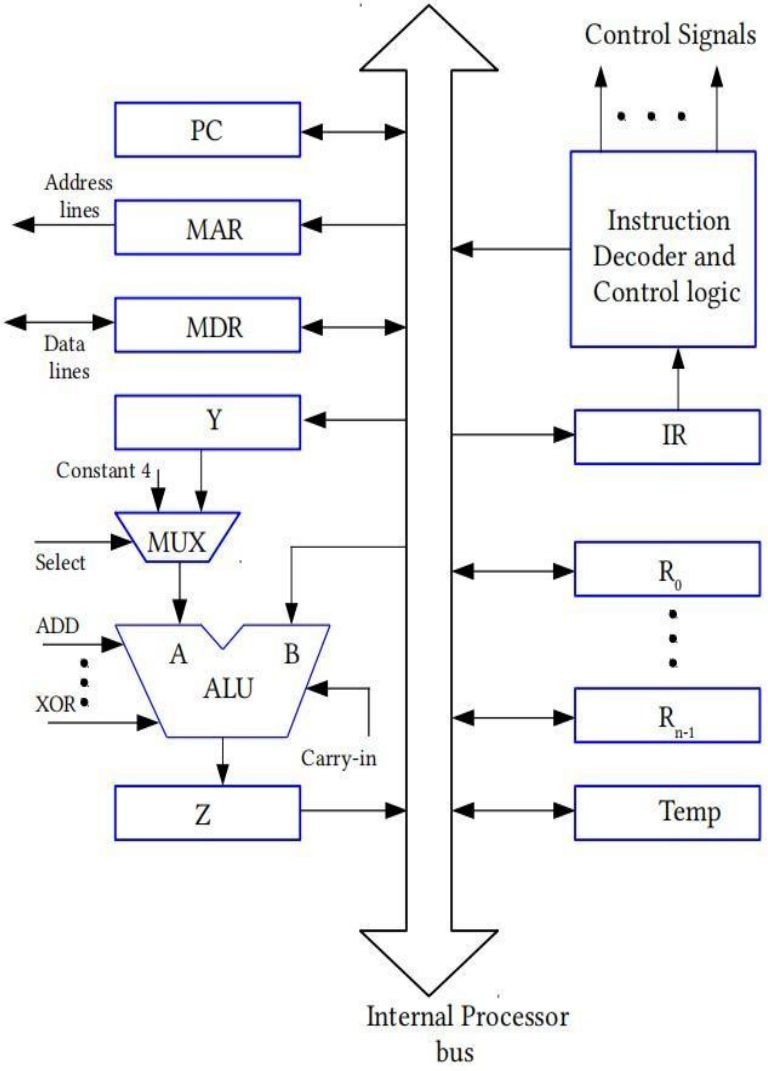
1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R1out, Select4, Sub, Zin
5. Zout, R1in, MARin, Read
6. R2out, Yin, WMFC
7. MDRout, SelectY, MUL,Zin
8. Zout, R2in, end



Execution of a Complete Instruction : Autoincrement mode

MUL (R1)+, R2

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R1out, MARin, Read, Select4, Add, Zin
5. Zout, R1in
6. R2out, Yin, WMFC
7. MDRout, SelectY, Mul, Zin
8. Zout, R2in, end



MultiBus CPU Organization

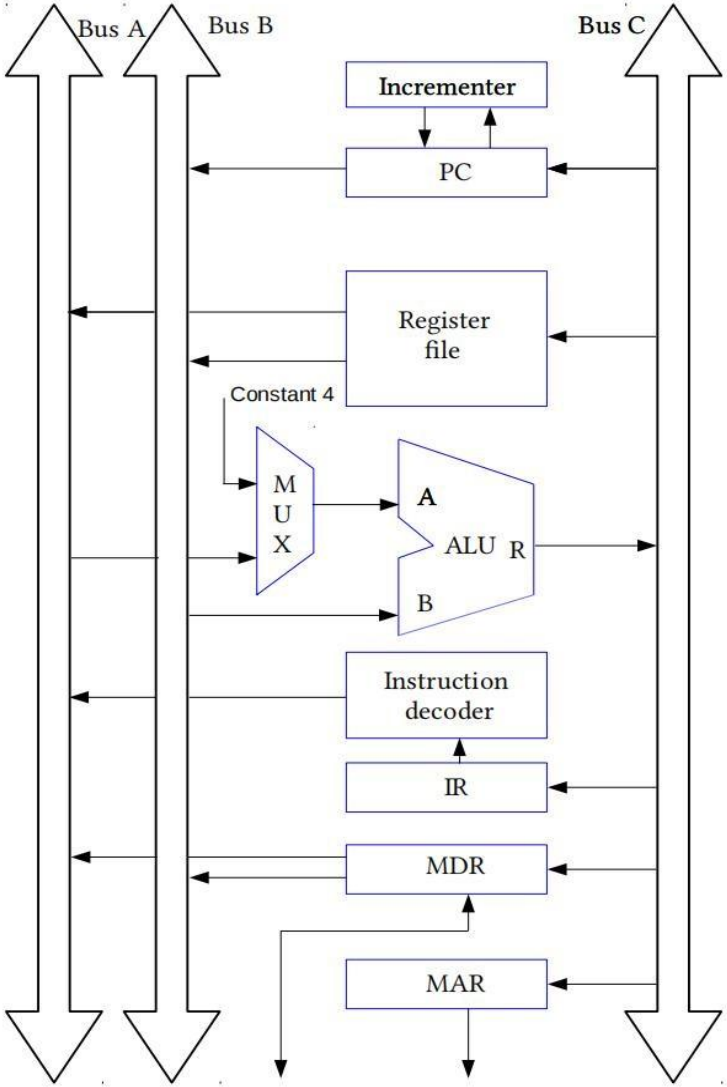
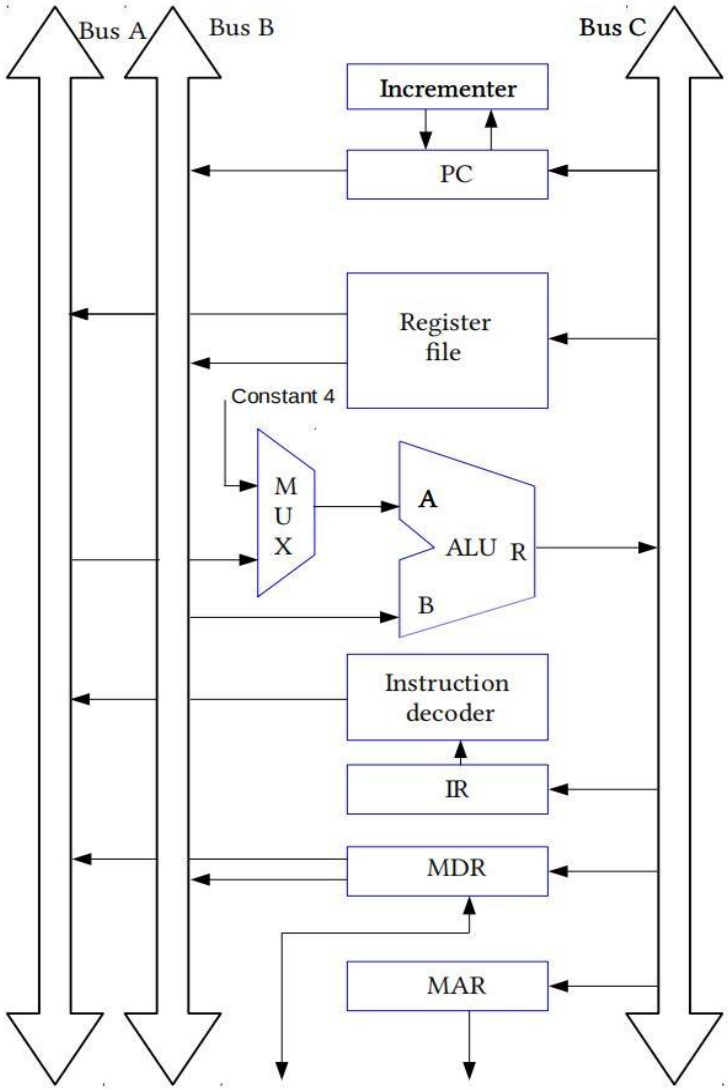


Fig. Three Bus CPU Organization

MultiBus CPU Organization

Add R4, R5, R6

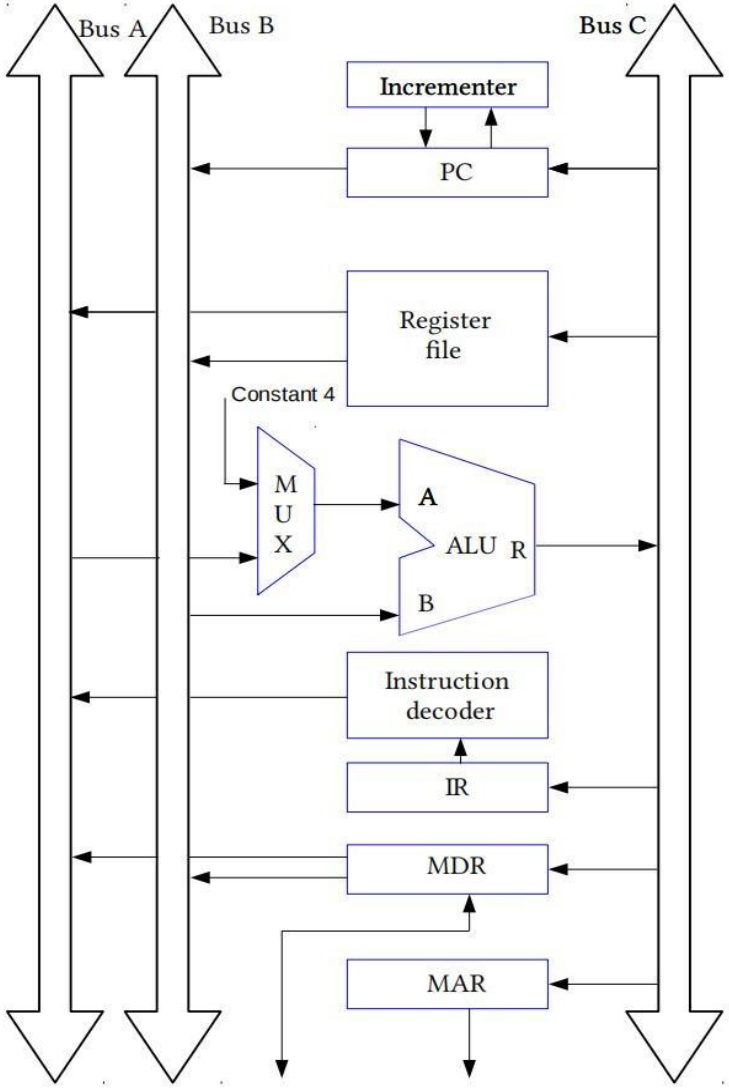
1. PC_{out}, R=B, MAR_{in}, Read, IncPC
2. WMFC
3. MDR_{outB}, R=B, IR_{in}
4. $R4_{outA}$, $R5_{outB}$, SelectA, ADD, $R6_{in}$, end



MultiBus CPU Organization

Add (R4), R5

1. PC_{out}, R=B, MAR_{in}, Read, IncPC
2. WMFC
3. MDR_{outB}, R=B, IR_{in}
4. $R4_{outB}$, R=B, MAR_{in}, Read
5. WMFC
6. MDR_{outB}, $R5_{outA}$, SelectA, ADD, $R5_{in}$, end



Hardwired Control Unit

To execute instructions, the processor must have some means of generating the control signals needed in the proper sequence.

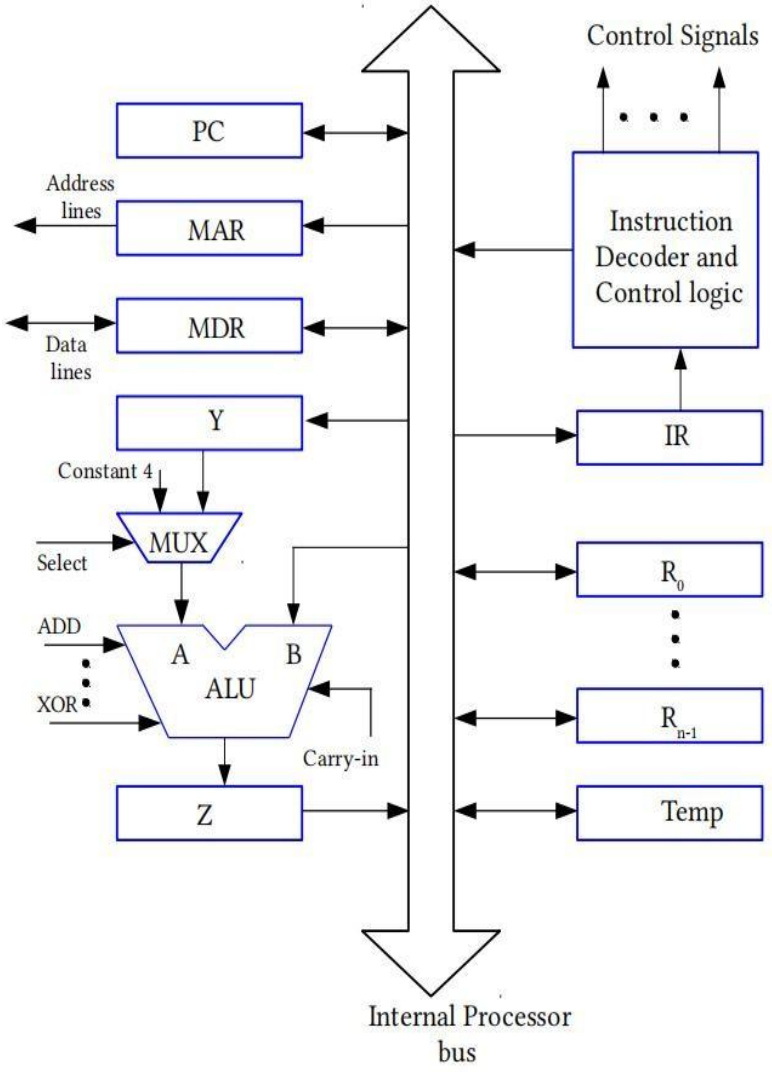
Two categories:
Hardwired control Unit and
Microprogrammed control Unit

Hardwired system can operate at high speed; but with little flexibility.

Execution of a Complete Instruction

Add (R3), R1

1. PCout, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, SelectY, Add, Zin
7. Zout, R1in, end



Hardwired Control Unit

This type of CU is designed using a nmber of combinational and sequential circuits like gates, FFs, decoder, encoder and other digital circuis.

For CU to perform it's function, it has some **inputs** that allows it to determine the state of the system and outputs that allows it to control the behaviour of the system.

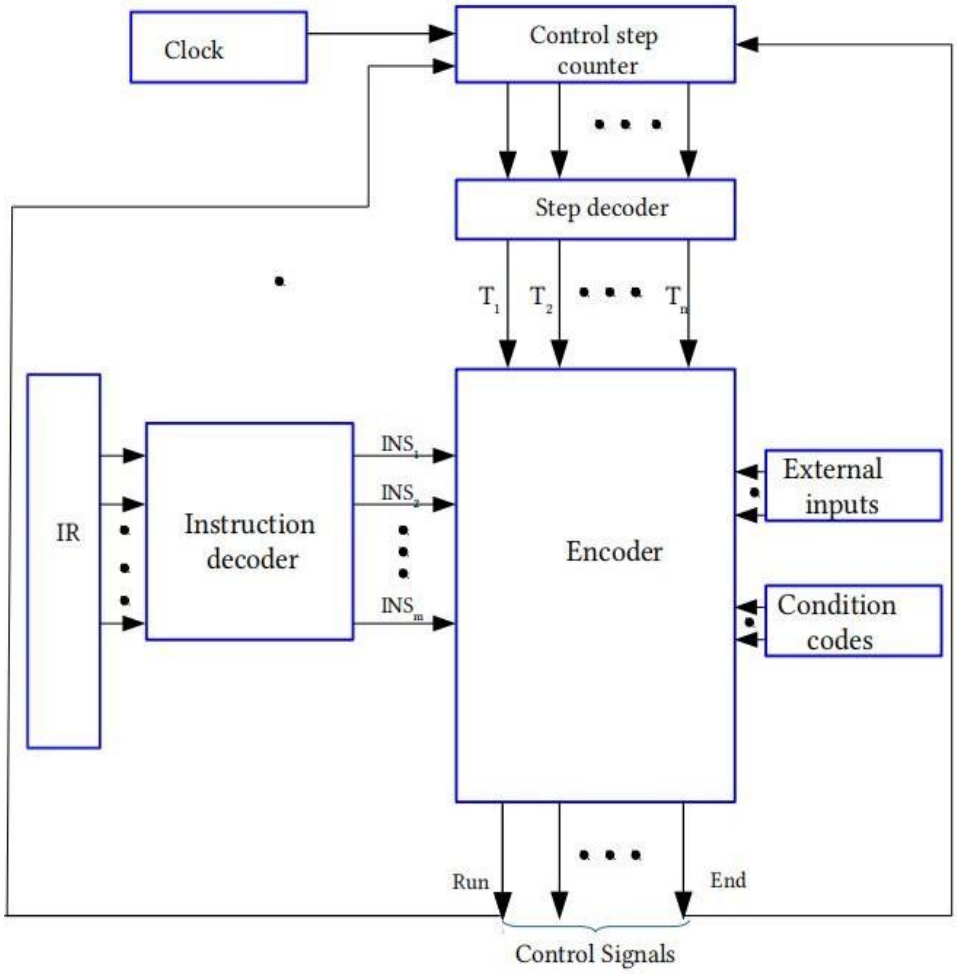
These inputs are the **external specification** of the CU.

Internally, the CU must have some logic to perform its sequencing and execution function.

Hardwired Control Unit

Inputs to CU:

1. Clock(Contents of control step counter)
2. Instruction Register (Contents of IR)
3. Flags(Contents of condition codes)
4. Control Signals from External Bus (e.g., MFC)



Hardwired Control Unit

Steps used in generating a control signal: (e.g., Zin)

1. Find out the control sequence for the instructions supported by the ISA.

2. Next find out , in which instructions the signal is appearing and then find out the step number of **that instruction** the signal is appearing.

3.Say, Zin is appearing in the **6th step** of the instruction**Add (R3), R1.** It means Zin signal need to be generated for the step no 6 of ADD instruction, so when **both** the cases are true, Zin need to be generated, like that for **JMP L1 instruction ,** Zin is generated in step no 4. i.e., in **either of the two** instructions Zin signal need to be generated.

4. So, the logic function, for Zin will be OR of the above two AND cases.
Zin= ADD.T6+ JMP.T4+.....[+.....indicates other possible cases]

5.Again, we hve seen that Zin is required for all the instructions in the step no 1 during the fetch of any instruction i.e., irrespective of any instruction, in the step no 1 Zin required. So,
Zin= T1+ ADD.T6+ JMP.T4+.....

Generating the Zin Signal

Add (R3), R1

1. PC out, MARin, Read, Select4, Add, **Zin**
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, SelectY, Add, **Zin**
7. Zout, R1in, end

JMP L1

1. PC out, MARin, Read, Select4, Add, **Zin**
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address field of_IRout, SelectY, Add, **Zin**
5. Zout, PCin, end

Generating the Zin Signal

BR <0 L1

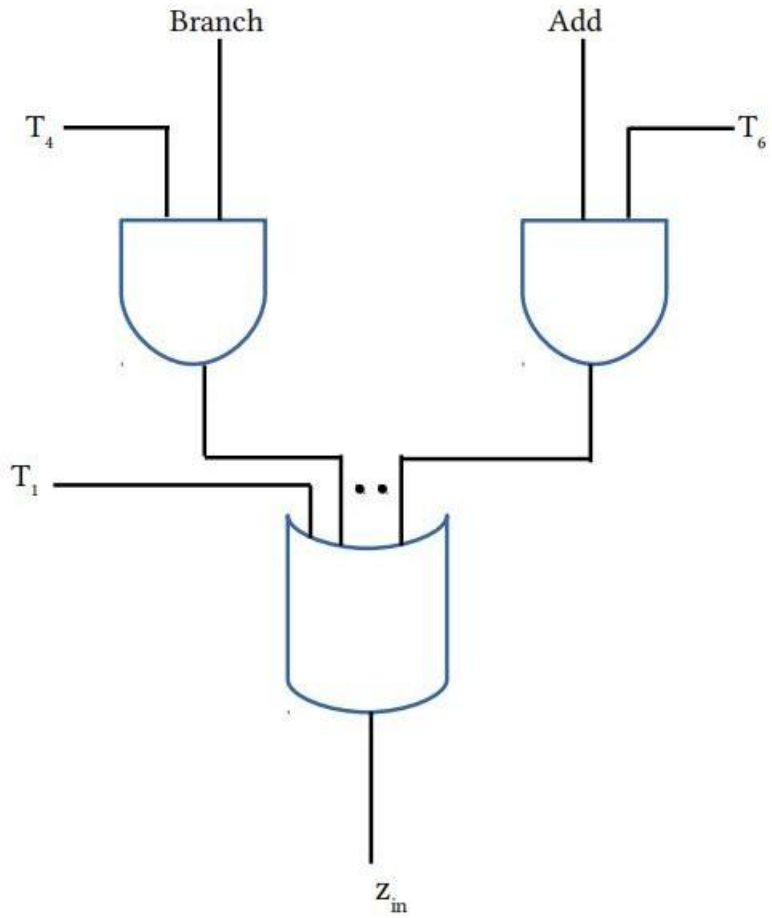
1. PC out, MARin, Read, Select4, Add, **Zin**
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. Address_field_of_IRout, SelectY, Add, **Zin,** if N==0,then end
5. Zout, PCin, end

Step 2: We have seen that Zin is active for all the instructions in control sequence no 1. i.e., it is not dependent on the instruction.
Next, Zin is active in the step **no 6** for the **ADD** instruction,
Zin is active in the step **no 4** for the **JMP** instruction,
Zin is active in the step **no 4** for the **BR** instruction.....

Step 3:

Zin= T1 + ADD.T6 + JMP.T4 + BRN.T4.N

Logic function for Zin Signal



Logic function for End signal

End= ADD.T7 + BRN.T5.N + BRN.T4.$\overline{N}$ + BR.T5+.....

Problem

A hardwired CPU uses 10 control signals S1 to S10, in various time steps T1 to T5, to implement 4 instructions I1to I4 as shown below:

	T1	T2	T3	T4	T5
I1	S1, S3, S5	S2, S4, S6	S1, S7	S10	S3,S8
I2	S1, S3, S5	S8,S9, S10	S5, S6, S7	S6	S10
I3	S1, S3, S5	S2, S6, S7	S2, S6, S9	S10	S1, S3
I4	S1, S3, S5	S2, S6, S7	S5, S10	S6, S9	S10

Write the expressions to represent the circuit for generating control signals S5 and S10 respectively? What will be the specification of step decoder and instruction decoder in the hardwired control unit?

Solution

Step 1: Find the required signals S5 and S10 in the given control seuence.

	T1	T2	T3	T4	T5
I1	S1, S3, **S5**	S2, S4, S6	S1, S7	**S10**	S3,S8
I2	S1, S3, **S5**	S8,S9, **S10**	**S5**, S6, S7	S6	**S10**
I3	S1, S3, **S5**	S2, S6, S7	S2, S6, S9	**S10**	S1, S3
I4	S1, S3, **S5**	S2, S6, S7	**S5**, **S10**	S6, S9	**S10**

$S5 = I1 + I2.T3 + I4.\ T3$

$S10 = I1.T4 + I2.(T2 + T5) + I3.T4 + I4(T3 + T5)$

Problem

Write the expressions to represent the circuit for generating control signals S5 and S10 respectively? What will be the specification of step decoder and instruction decoder in the hardwired control unit?

Step decoder 3: 8

Instruction decoder 2 : 4

Problem

A computer has 58 instructions; each instruction requires at most 15 steps to complete its execution. What will be the specification of instruction and step counter decoder used in hardware control unit design?

Step decoder 4: 16

Instruction decoder 6 : 64

Problem

A hardwired CPU has only 3 instructions I1, I2 and I3, which use the following signals in time steps T1-T5

	T1	T2	T3	T4	T5
I1	Ain,Bout, Cin	PCout,Bin	Zout,Ain	Bin,Cout	End
I2	Cin,Bout, Din	Aout,Bin	Zout,Ain	Bin,Cout	End
I3	Din,Aout	Ain,Bout	Zout,Ain	Dout,Ain	End

Write the logic function for generating the signal Ain?

Ain= I1.T1 + T3 + I3(T2+T4)

Consider an example of memory organization as shown in the figure below. Which valuewill be loaded into the accumulator when the instruction “LOAD DIRECT 3” is executed

ML Address	0	1	2	3	4	5	6	7
Content	10	23	25	20	12	3	1	2

3

25

12

20

Consider an example of memory organization as shown in the figure below. Which valuewill be loaded into the accumulator when the instruction “LOAD INDIRECT 7” is executed

ML Address	0	1	2	3	4	5	6	7
Content	10	23	25	20	12	3	1	2

2

25

7

20

Consider a three word machine instruction-

ADD A[R0], @B

The first operand (destination) “A[R0]” uses indexed addressing mode with R0 as the index register. The second operand operand (source) “@B” uses indirect addressing mode. A and B are memory addresses residing at the second and the third words, respectively. The first word of the instruction specifies the opcode, the index register designation and the source and destination addressing modes. During execution of ADD instruction, the two operands are added and stored in the destination (first operand).

The number of memory cycles needed during the execution cycle of the instruction is??

Microprogrammed Control Unit

An alternative to a hardwired control unit is a microprogrammed control unit, in which the logic of the control unit is specified by a microprogram.

A microprogram consists of a sequence of instructions in a microprogramming language. These are very simple instructions that specify micro-operations.

The term microprogram was first coined by M. V. Wilkes in the early 1950s

Introduction to Microprogrammed CU

Add (R3), R1

1. PC out, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, SelectY, Add, Zin
7. Zout, R1in, end

µinstruction	PCout	MARin	MDRout	Read	Select	Add	Zin	Zout		Yin	WMFC	IRin	R3out	R1out	R1in	R2in	R2out	Sub	XOR	MUL	Div	Rin	R4in	R4out	Write	.	.	.	.	End
1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Terms Related to Microprogrammed Control Unit

Microinstruction (Control Word): For each microoperation the CU generates a set of control signals. Thus for any microoperation, each control line emanating from CU is either on or off . This condition can be represented by a binary digit for each control line. Each row of the previous table represents a microinstruction. i.e., in a microinstruction, each bit position is fixed for a particular control signal, if a signal is generated in that particular micro-instruction, then that bit postion value will be 1 else it will be 0.

Microroutine: A sequence of control words corresponding to a particular instruction is called as microroutine for that instruction. The previous table represents the microroutine for the instruction **Add (R3), R1**

Control Store: The instruction set of any computer is finite. The microroutines for all the instructions in the instruction set is stored in a special memory called as Control store/ Control Memory.

μPC: points to the next microinstruction that need to be fetched from Control store.

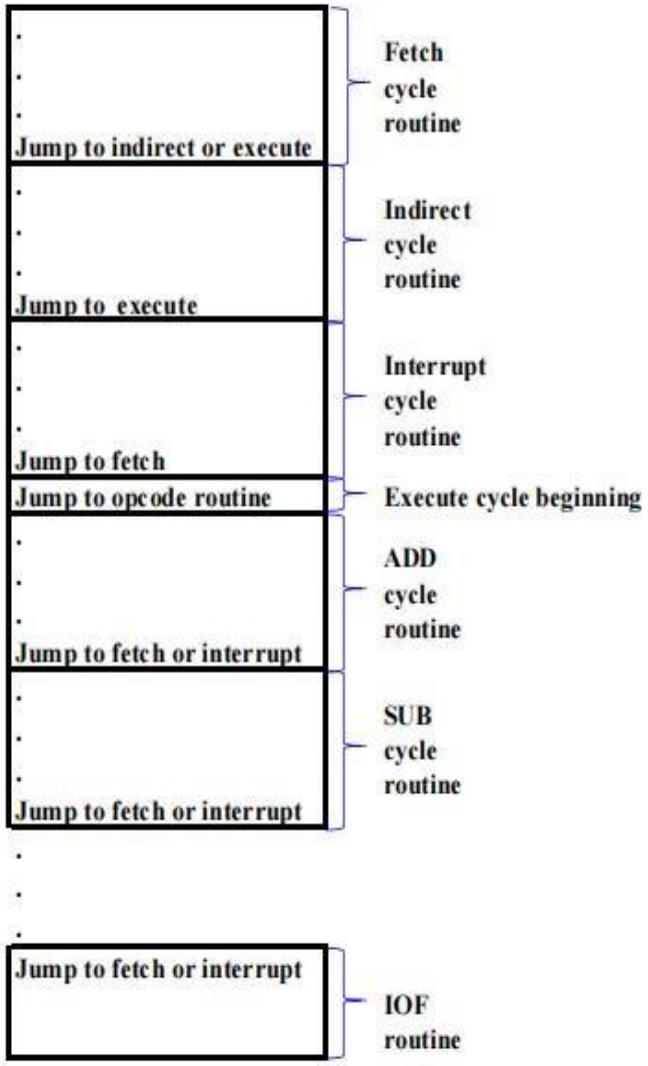
Organization of Control Memory

The fig. shows how the control words or microinstructions could be arranged in a control memory.

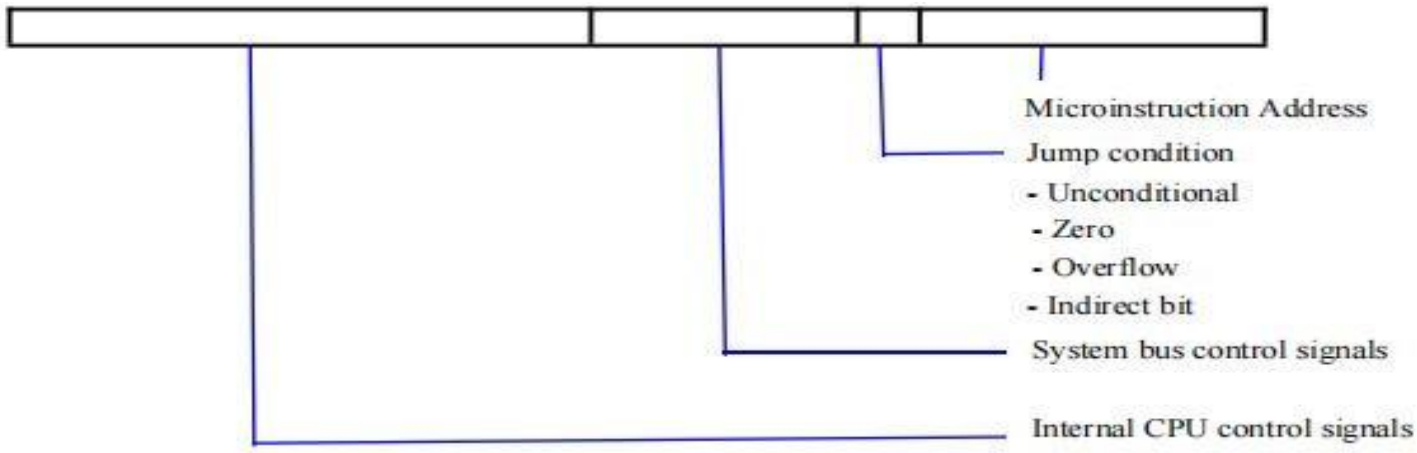
The microinstructions in each routine are to be executed sequentially.

Each routine ends with a branch or jump instruction indicating where to go next.

There is a special execute cycle routine whose only purpose is to signify that one of the machine instruction routines (AND, ADD, and so on) is to be executed next, depending on the current opcode.



Typical Microinstruction Format



There is one bit for each internal processor control line and one bit for each system bus control line.

There is a condition field indicating the condition under which there should be a branch, and there is a field with the address of the microinstruction to be executed next when a branch is taken.

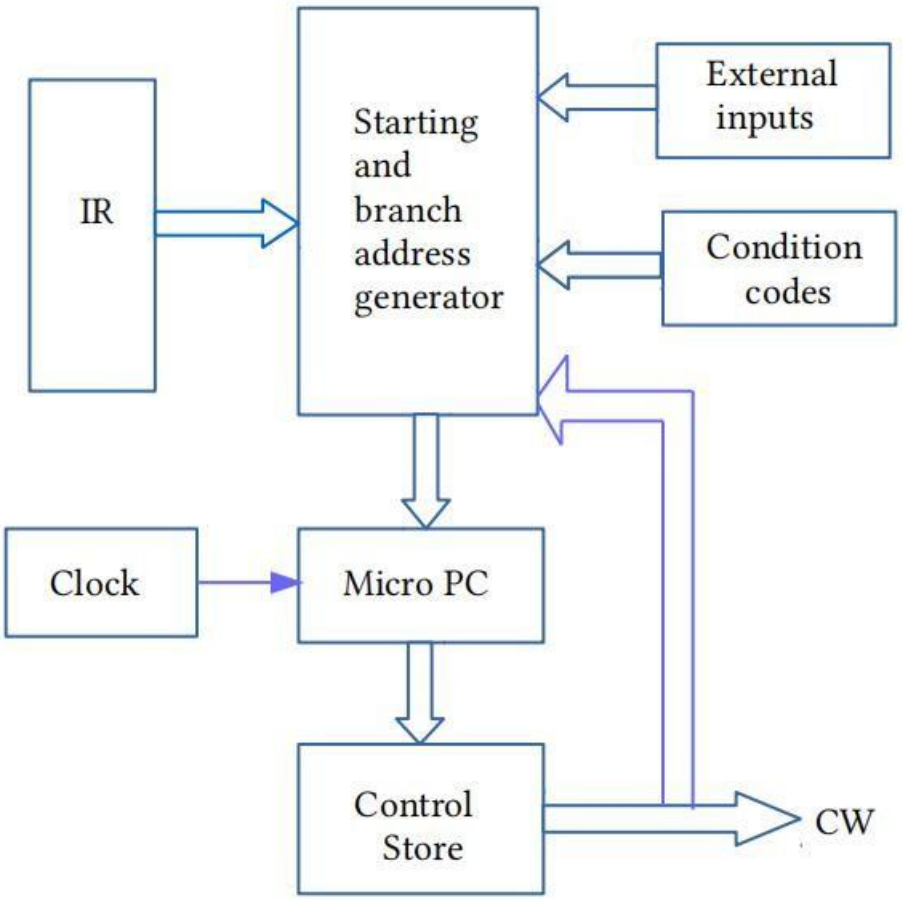
Interpretation

To execute this microinstruction, turn on all the control lines indicated by a 1 bit; leave off all control lines indicated by a 0 bit. The resulting control signals will cause one or more micro-operations to be performed.

If the condition indicated by the condition bits is false, execute the next microinstruction in sequence.

If the condition indicated by the condition bits is true, the next microinstruction to be executed is indicated in the address field.

Microprogrammed CU



To execute any instruction, the CU should first find the starting address of the and then can generate the control signals in sequence by reading the control words one by one.

Microprogrammed CU

Address	Microinstruction
0	PC out, MARin, Read, Select4, Add, Zin
1	Zout, PCin, Yin, WMFC
2	MDRout, IRin
3	Branch to starting address of appropriate microroutine
.....	.....
30	If N==0, then branch to microinstruction 0
31	Offset_field_of_IRout, SelectY, Add, Zin
32	Zout, PCin, end

Fig. Microroutind for Branch<0 instruction

Microprogrammed Control Unit

In this control unit, the μPC is incremented every time a new microinstruction is fetched from the microprogram memory, except in the following situations:

1. When a new instruction is loaded into the IR, the μPC is loaded with the starting address of the microroutine for that instruction.

2. When a Branch microinstruction is encountered and the branch condition is satisfied, the μPC is loaded with the branch address.

3. When an End microinstruction is encountered the μPC is loaded with the address of the first CW in the microroutine for the instruction fetch cycle.

Thank You