COA

Ch2- Part2
Introduction to CPU Organization

By

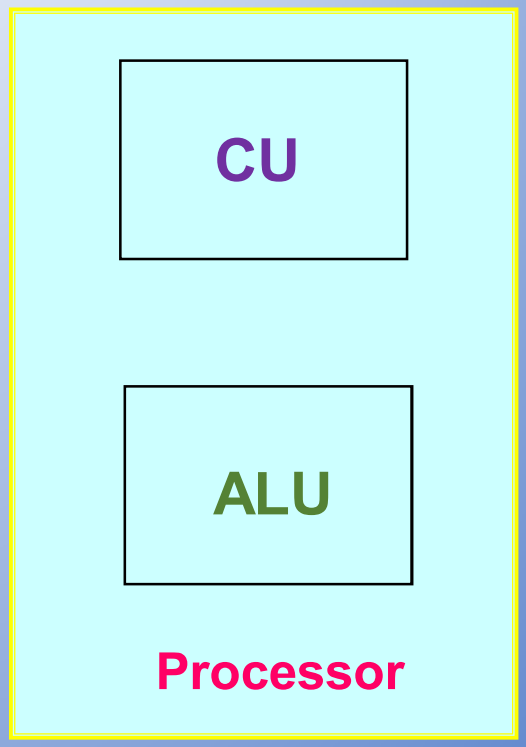
Dr. Raghad Samir Al Najim

2018@SPU

CPU Organization

CPU contains two main units:

ALU and Control unit



ALU

The arithmetic logic unit (ALU) carries out the logic operations (such as comparisons) and arithmetic operations (such as add or subtract) required during the program execution.

Generally an ALU has two data inputs and one data output.

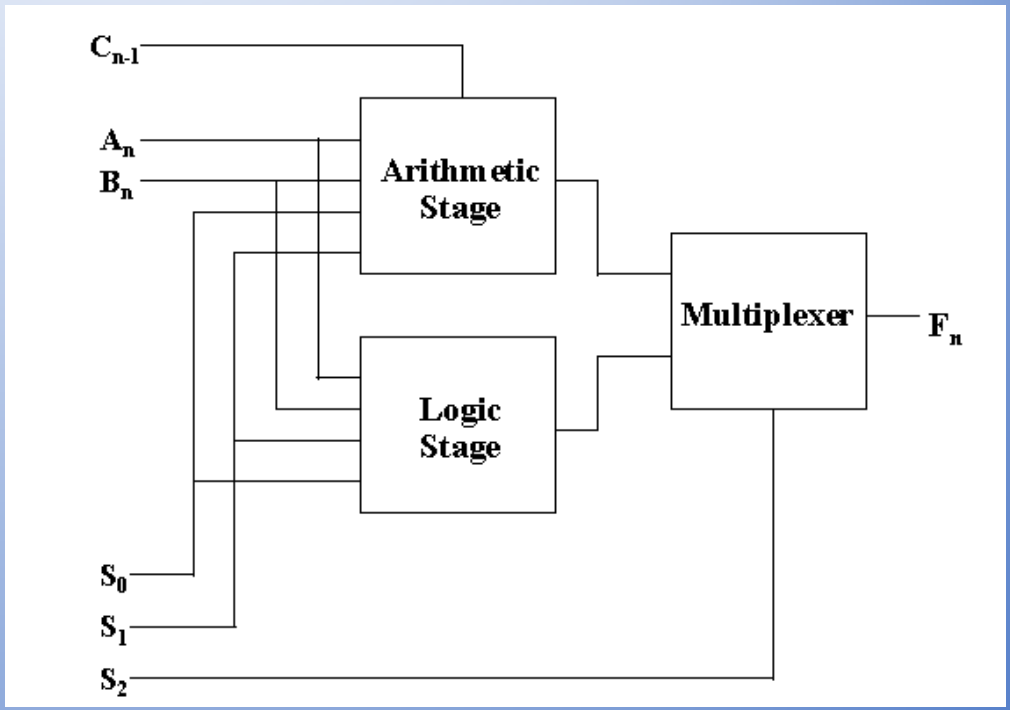
Operations performed in the ALU affect bits in the status register(Flag register).

The ALU knows which operations to perform because it is controlled by signals from the control unit.

ALU Components

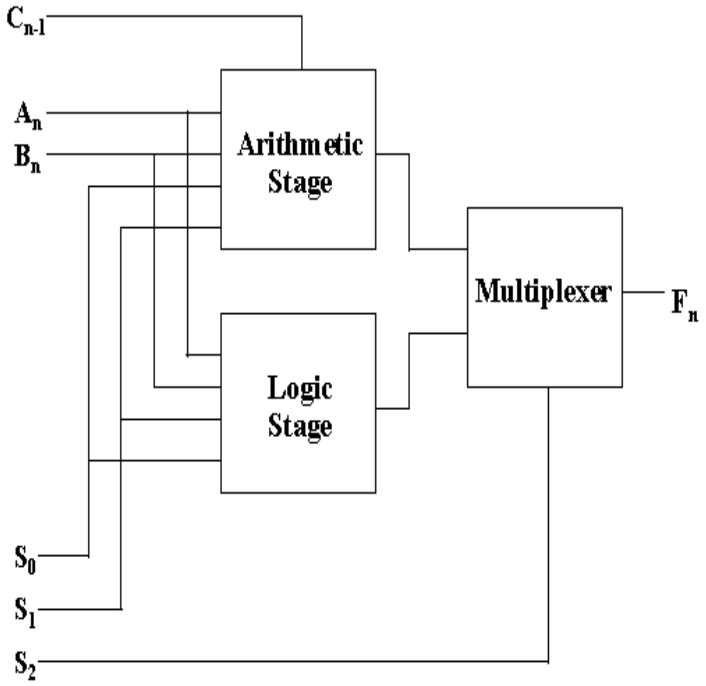
- Arithmetic
  - Parallel Add
  - One Full Adder per Bit
  - Selection Logic
- Logic
  - Gates
  - Multiplexer

Internal Structure of ALU

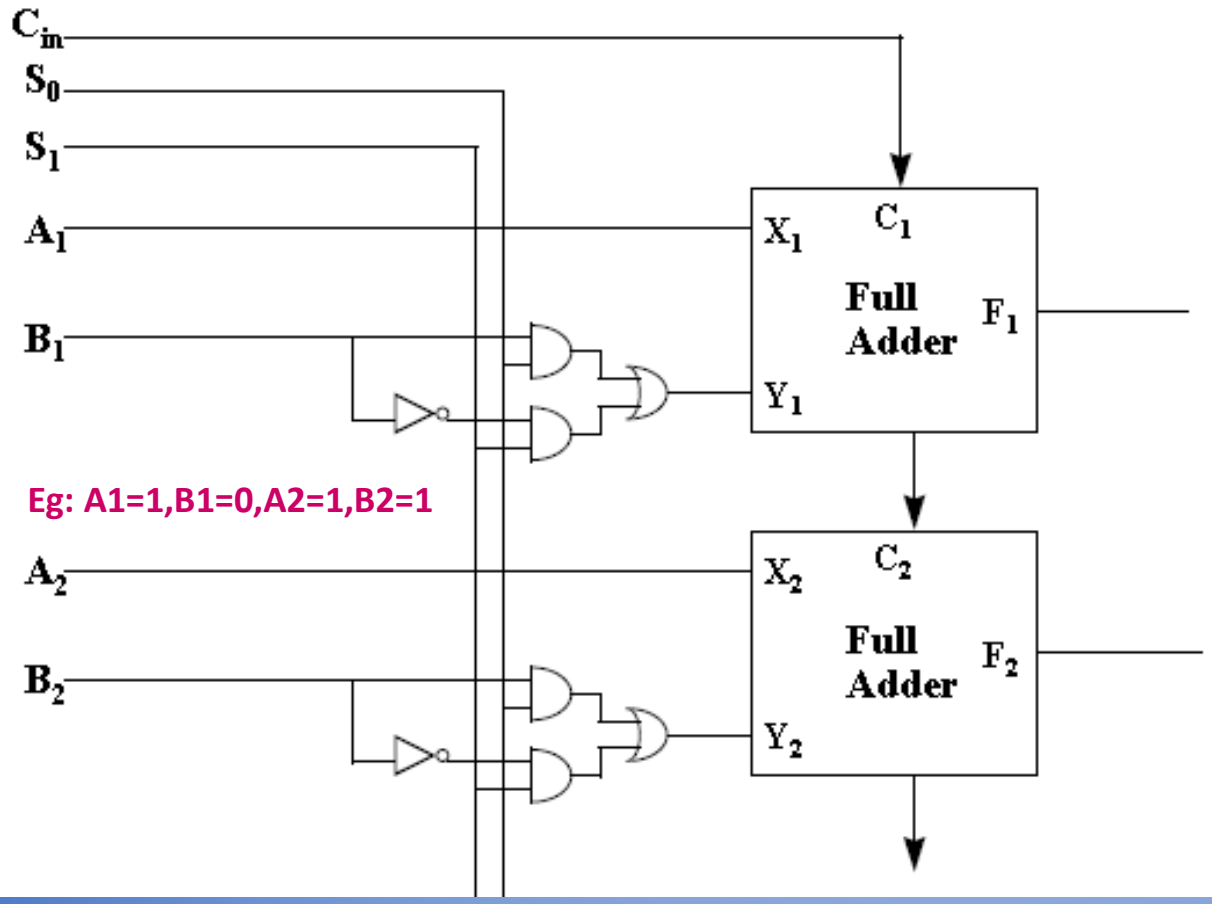


ALU Function Table

Operation Select S_1	S_0	S_b	C_{in}	Operation	Function
0	0	0	0	F=A	Transfer A
0	0	0	1	F=A+1	Increment A
0	0	1	0	F=A+B	Add A and B
0	0	1	1	F=A+B+1	Add A and B With Carry
0	1	0	0	F=A+B'	Add A and One's Compement of B
0	1	0	1	F=A+B'+1	Subtract B From A
0	1	1	0	F=A-1	Decrement A
0	1	1	1	F=A	Transfer A
1	0	0	0	F=AB	AND
1	0	1	0	F=A+B	OR
1	1	0	0	F=A XOR B	Exclusive OR
1	1	1	0	F=A'	Complement

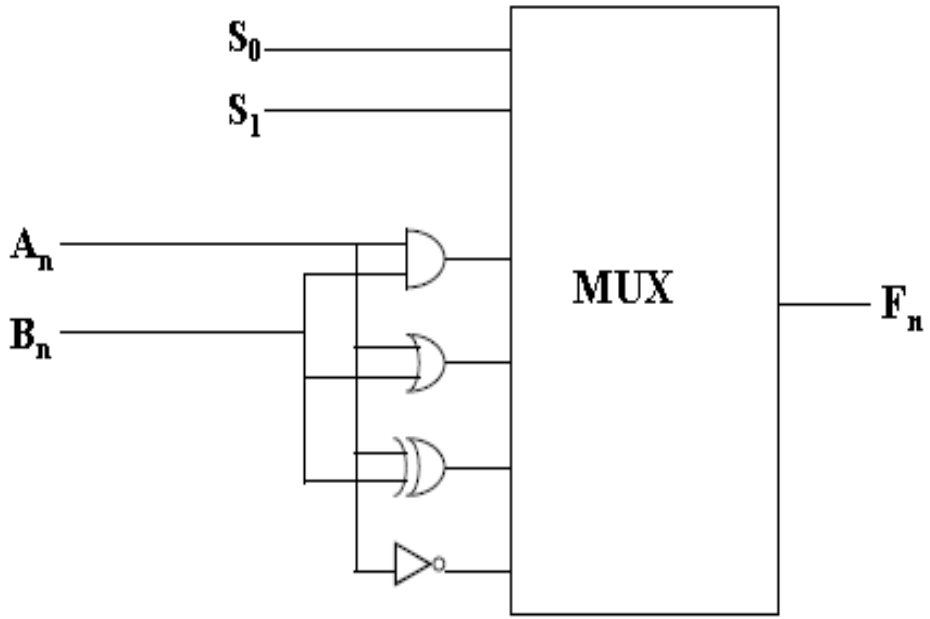


Circuit for Arithmetic Component



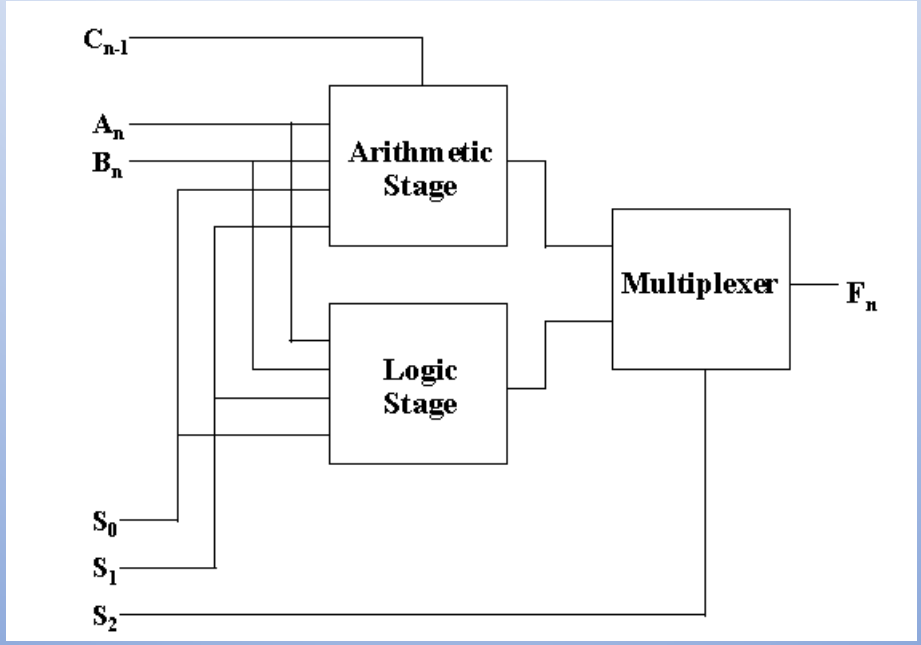
Operation Select				Function
S_1	S_1	S_0	C_{in}	
0	0	0	0	Transfer A
0	0	0	1	Increment A
0	0	1	0	Add A and B
0	0	1	1	Add A and B With Carry
0	1	0	0	Add A and One's Compement of B
0	1	0	1	Subtract B From A
0	1	1	0	Decrement A
0	1	1	1	Transfer A
1	0	0	0	AND
1	0	1	0	OR
1	1	0	0	Exclusive OR
1	1	1	0	Complement

Circuit for Logic Component



Operation Select S_1	S_0	S_0	C_{in}	Function
0	0	0	0	Transfer A
0	0	0	1	Increment A
0	0	1	0	Add A and B
0	0	1	1	Add A and B With Carry
0	1	0	0	Add A and One's Compement of B
0	1	0	1	Subtract B From A
0	1	1	0	Decrement A
0	1	1	1	Transfer A
1	0	0	0	AND
1	0	1	0	OR
1	1	0	0	Exclusive OR
1	1	1	0	Complement

Internal Structure of ALU



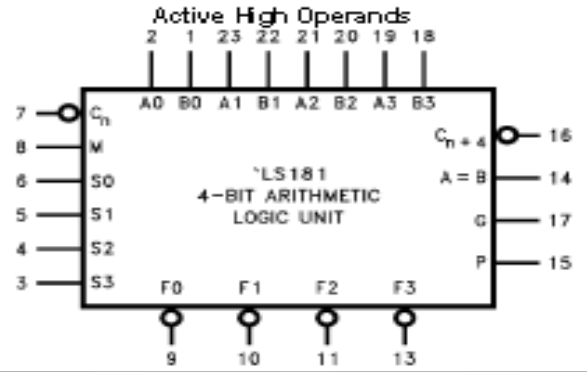
Pin Descriptions

Pin Names	Description
$\overline{A0}$–$\overline{A3}$	Operand Inputs (Active LOW)
$\overline{B0}$–$\overline{B3}$	Operand Inputs (Active LOW)
S0–S3	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\overline{F0}$–$\overline{F3}$	Function Outputs (Active LOW)
A = B	Comparator Output
$\overline{G}$	Carry Generate Output (Active LOW)
$\overline{P}$	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output

Function Table

Mode Select Inputs				Active LOW Operands & F_n Outputs		Active HIGH Operands & F_n Outputs	
				Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
S3	S2	S1	S0	(M = H)	(M = L)(C_n = L)	(M = H)	(M = L)(C_n = H)
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\overline{A}+B$	A + B
L	L	H	L	$\overline{A}+\overline{B}$	A$\overline{B}$ minus 1	$\overline{A}B$	A + $\overline{B}$
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\overline{A}+\overline{B}$	A plus (A + $\overline{B}$)	$\overline{AB}$	A plus A$\overline{B}$
L	H	L	H	$\overline{B}$	AB plus (A + $\overline{B}$)	$\overline{B}$	(A + B) plus A$\overline{B}$
L	H	H	L	$\overline{A} \oplus \overline{B}$	A minus B minus 1	A⊕B	A minus B minus 1
L	H	H	H	A + $\overline{B}$	A + $\overline{B}$	A$\overline{B}$	AB minus 1
H	L	L	L	$\overline{A}$B	A plus (A + B)	$\overline{A}$+B	A plus AB
H	L	L	H	A⊕B	A plus B	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	A$\overline{B}$ plus (A + B)	B	(A + $\overline{B}$) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A(Note 1)	Logic 1	A plus A(Note 1)
H	H	L	H	A$\overline{B}$	AB plus A	A+ $\overline{B}$	(A + B) plus A
H	H	H	L	AB	A$\overline{B}$ minus A	A+ B	(A + $\overline{B}$) plus A
H	H	H	H	A	A	A	A minus 1

Logic Symbols

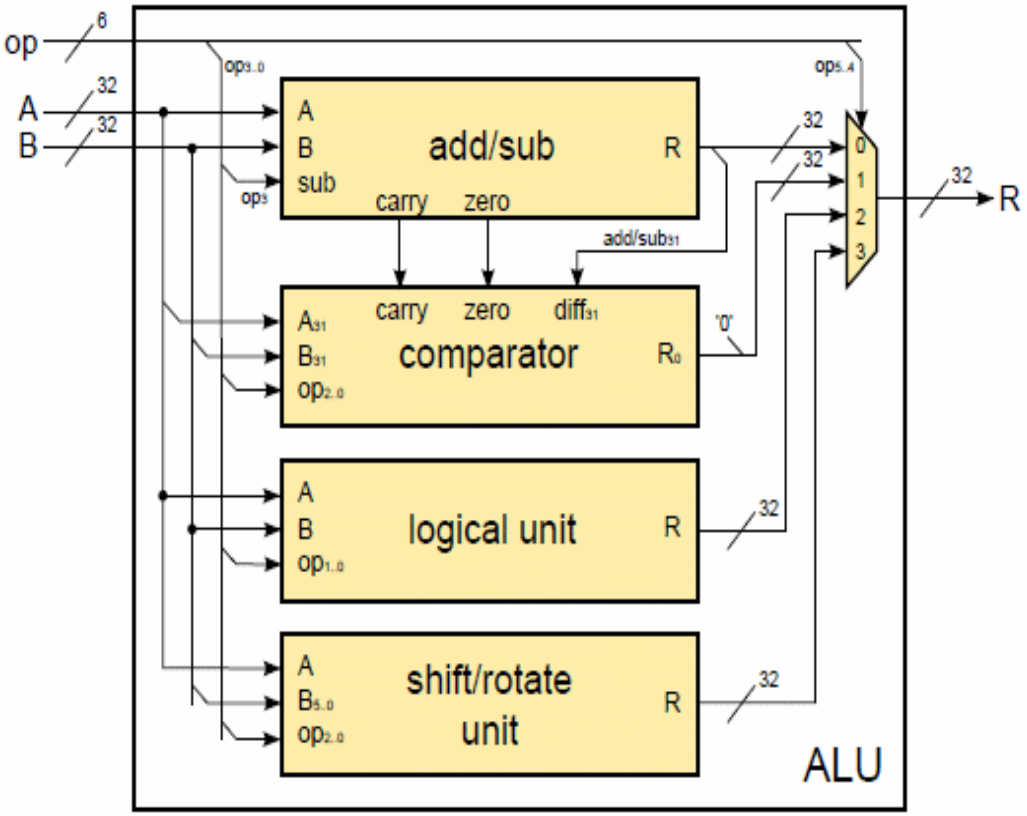


4- bits ALU

Complex ALU- 32 bits ALU

Operation	Operation Type	Opcode
$A + B$	Add/Sub	$000\phi\phi\phi$
$A - B$		$001\phi\phi\phi$
$A \geq B$ (signed)	Comparison	011001
$A < B$ (signed)		011010
$A \neq B$		011011
$A = B$		011100
$A \geq B$ (unsigned)		011101
$A < B$ (unsigned)		011110
A *nor* B	Logical	$10\phi\phi00$
A *and* B		$10\phi\phi01$
A *or* B		$10\phi\phi10$
A *xor* B		$10\phi\phi11$
A *rol* B	Shift/Rotate (Optional)	$11\phi000$
A *ror* B		$11\phi001$
A *sll* B		$11\phi010$
A *srl* B		$11\phi011$
A *sra* B		$11\phi111$

ϕ = *don't care*



Control Unit

The control unit is the “policeman” or “traffic manager” of the CPU. It monitors the execution of all instructions and the transfer of all information.

The control unit extracts instructions from memory, decodes these instructions, making sure data is in the right place at the right time, tells the ALU which registers to use, services interrupts, and turns on the correct circuitry in the ALU for the execution of the desired operation.

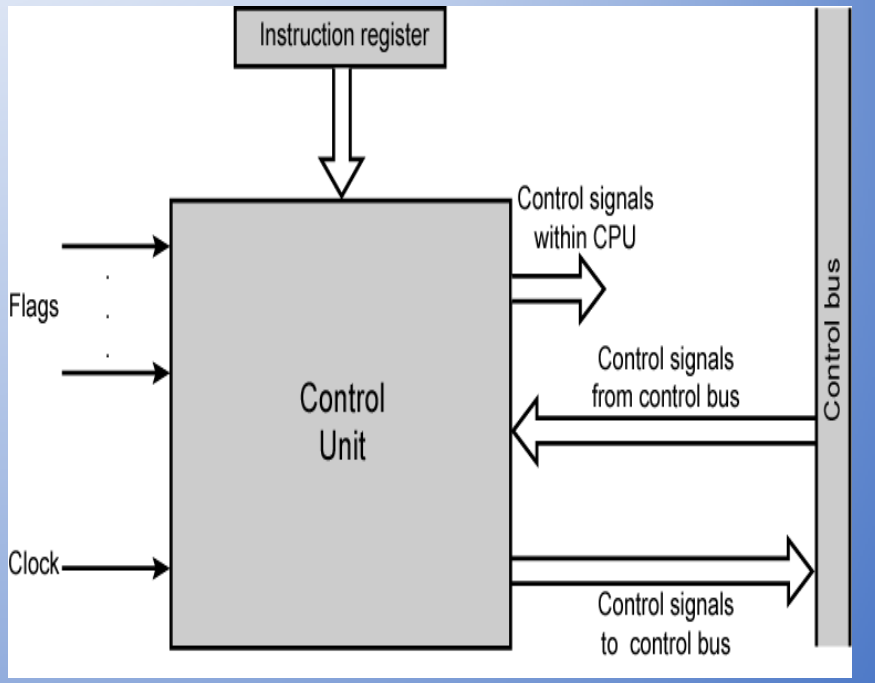
The control unit uses a program counter register to find the next instruction for execution and a status register to keep track of overflows, carries, borrows, …

Functions of Control Unit(CU)

- Sequencing
  - **CU causes the CPU to step through a series of micro-operations in proper sequence based on the program being executed**
- Execution
  - **CU causes each micro-operation to be performed**
- Control Signals
  - **External: inputs indicating the state of the system**
  - **Internal: logic required to perform the sequencing and execution functions**

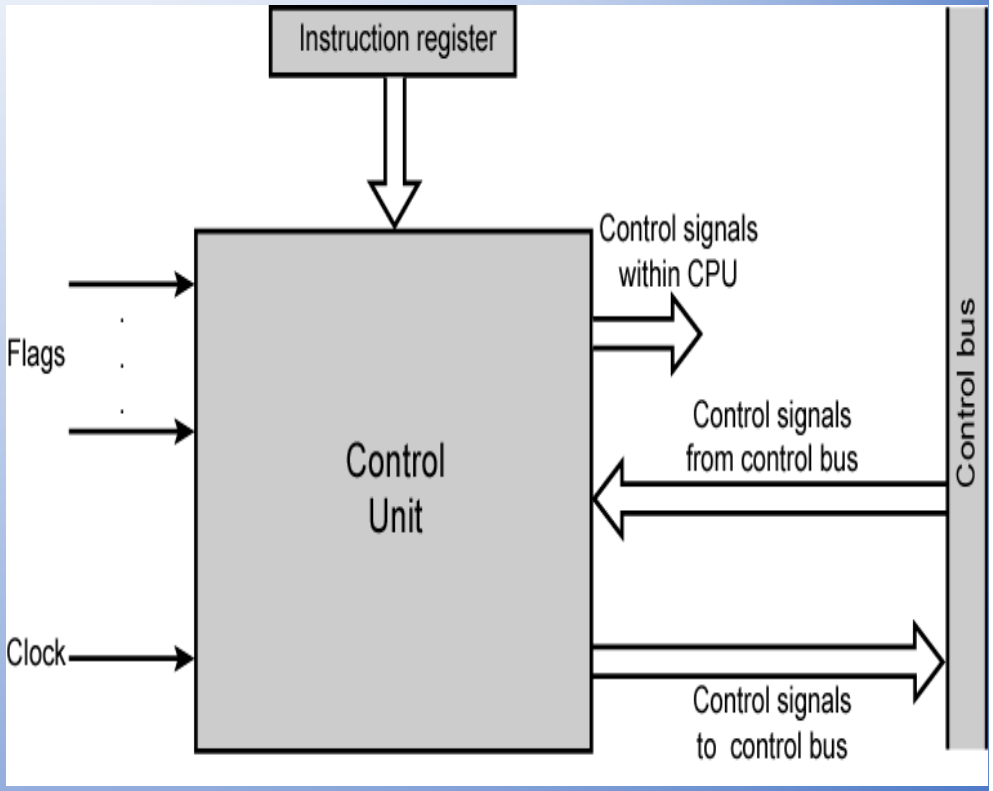
Control Signals- inputs & outputs

- Clock **(clock cycle time, processor cycle time)**
  - **One micro-instruction (or set of parallel micro-instructions) per clock cycle**
- Instruction register
  - **Opcode & addressing mode for current instruction**
  - **Determines which micro-instructions are performed**
- Flags – **Used to determine the**
  - **status of the CPU**
  - **Results of previous ALU operations**
- Signals from the control bus part of the system bus
  - **Interrupts**
  - **Acknowledgements**



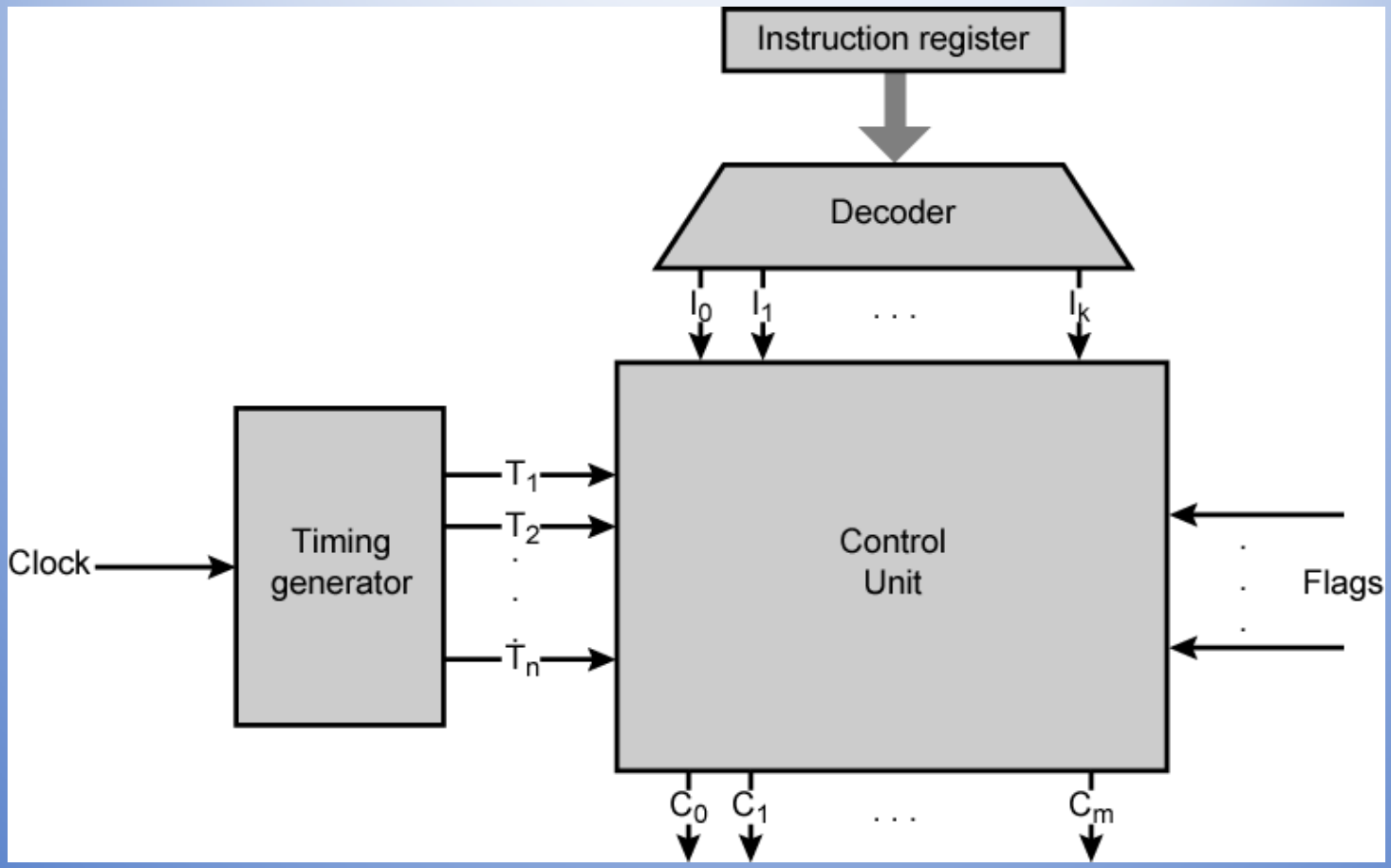
Control Signals- inputs & outputs

- Control Signals within CPU
  - **Cause data movement register to register**
  - **Activate specific ALU functions**
  - **Activate a specific data path**
- Control Signals to the control bus
  - **To memory via the system bus**
  - **To I/O modules**



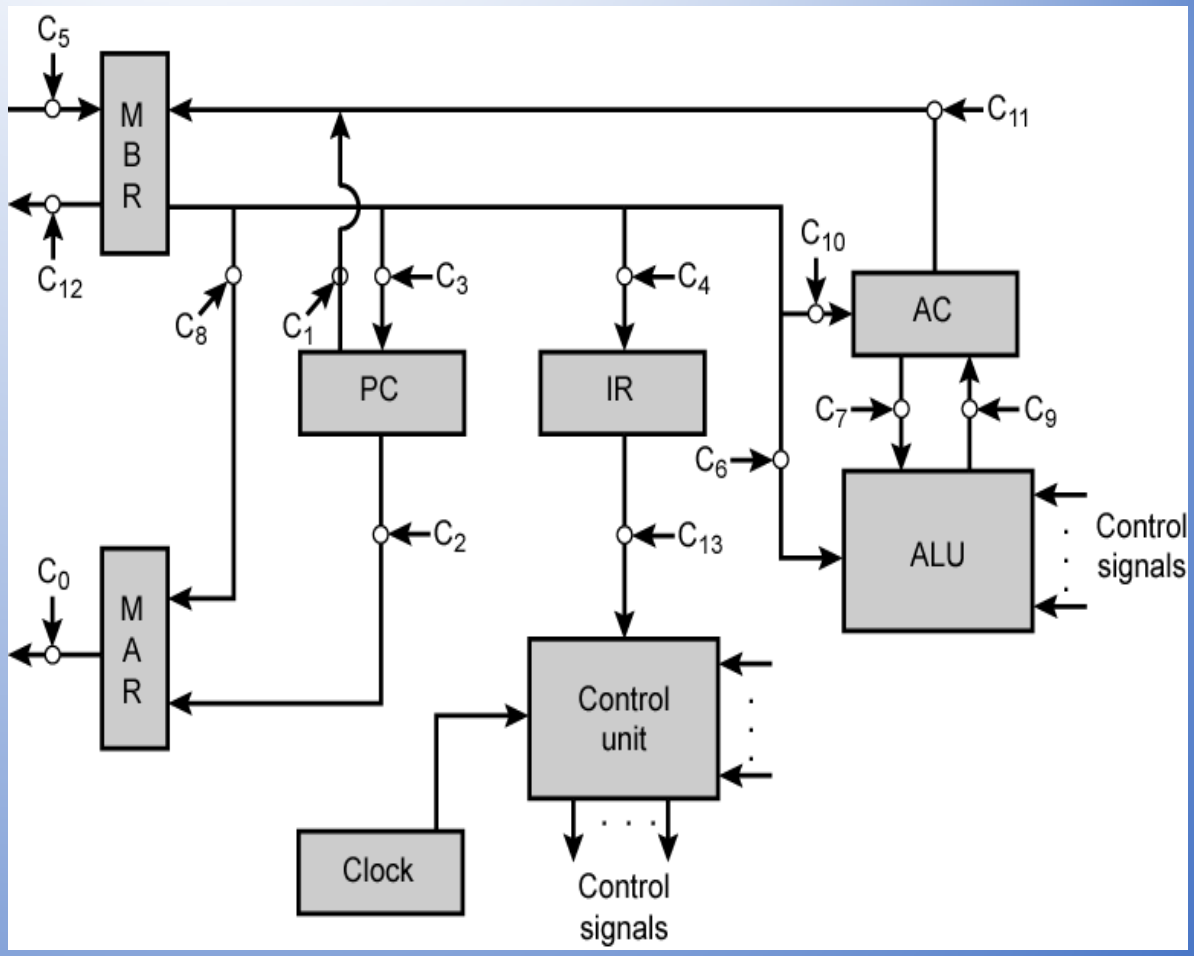
Model of Control Unit

Control Unit



Examples of Control Signal Sequence :

- MAR ← (PC)
  - **Control unit activates signal to open gates between PC and MAR**
- MBR ← M[MAR]
  - **Control unit opens the gates allowing contents of the MAR onto the address bus**
  - **Control unit sends a memory read control signal on the control bus**
  - **Control unit open the gates allowing contents of the data bus to be stored in the MBR**



Micro-sequencing of Instruction cycle

Micro-operation	Timing	Active control signal
Fetch	T1:MAR←PC	C2
	T2: MBR←M[MAR]: address PC←PC+1	C5, CR
	T3: IR←(MBR)	C4
Decoding	T1: MAR←IR[MAR]	C8
	T2: MBR←M[MAR]	C5,CR
CR= Read control signal to system bus