VLSI SYSTEMS AND ARCHITECTURE
2021-22
Lecture 10
Timing Signals

By Dr. Sanjay Vidhyadharan

Instruction Set Example

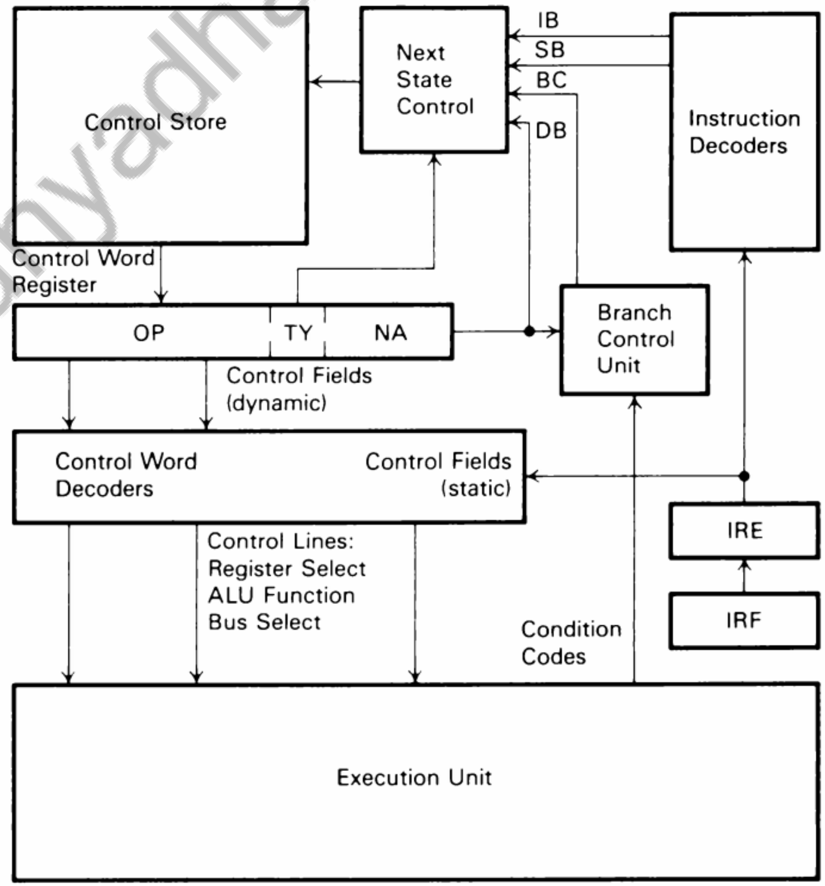
ADD

A R1,D2 (B2)

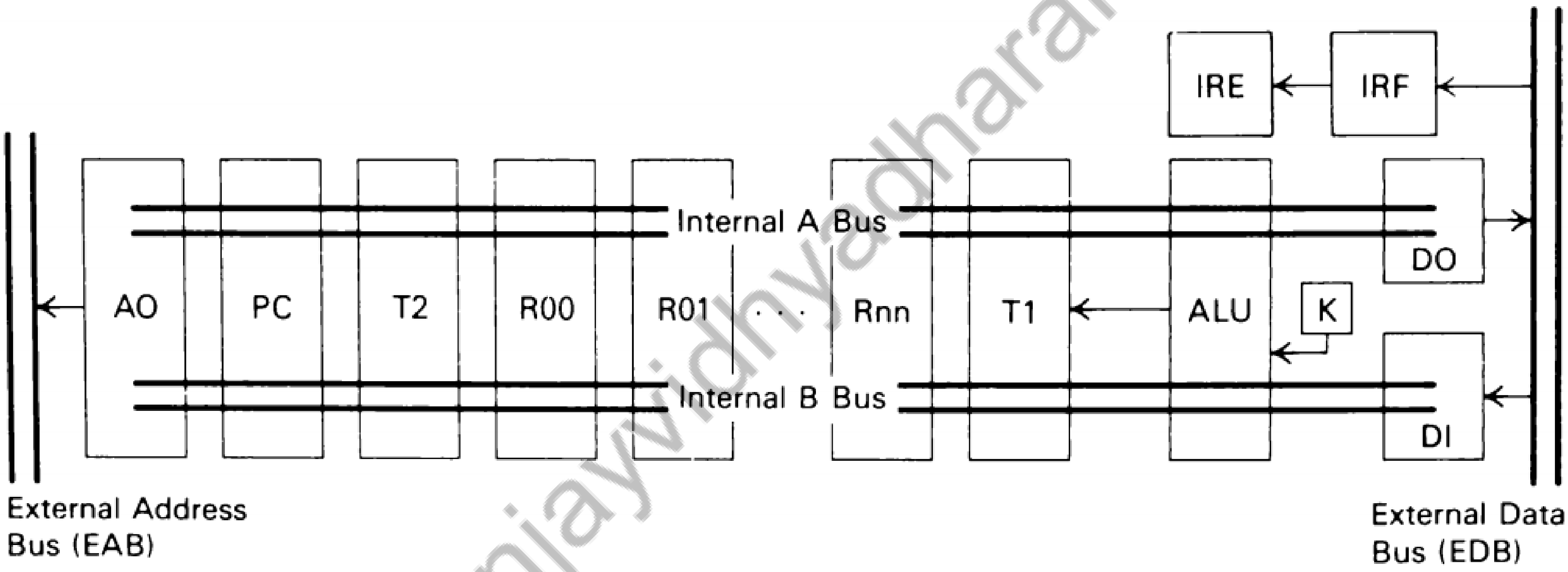
'5A'	R1	B2	D2
0	8	12	16 31

Here are the steps for the ADD instruction:

1. Fetch the first instruction halfword.
2. Find the ADD control word sequence.
3. Fetch the remaining instruction halfword.
4. Calculate the operand address.
5. Fetch the operand.
6. Add.
7. Store the answer.

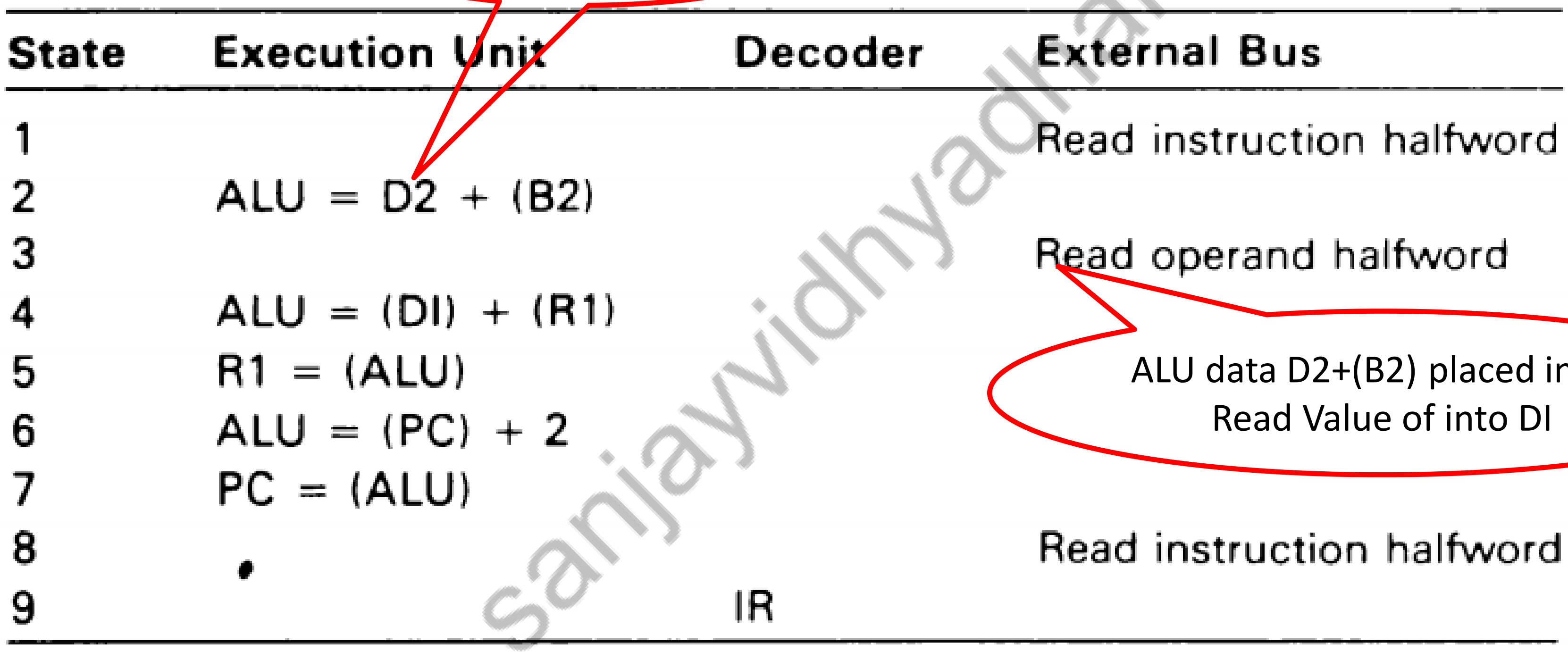


ADD Instruction Timing State



ADD Instruction Execution with No Overlap

ADD R1, D2 (B2)

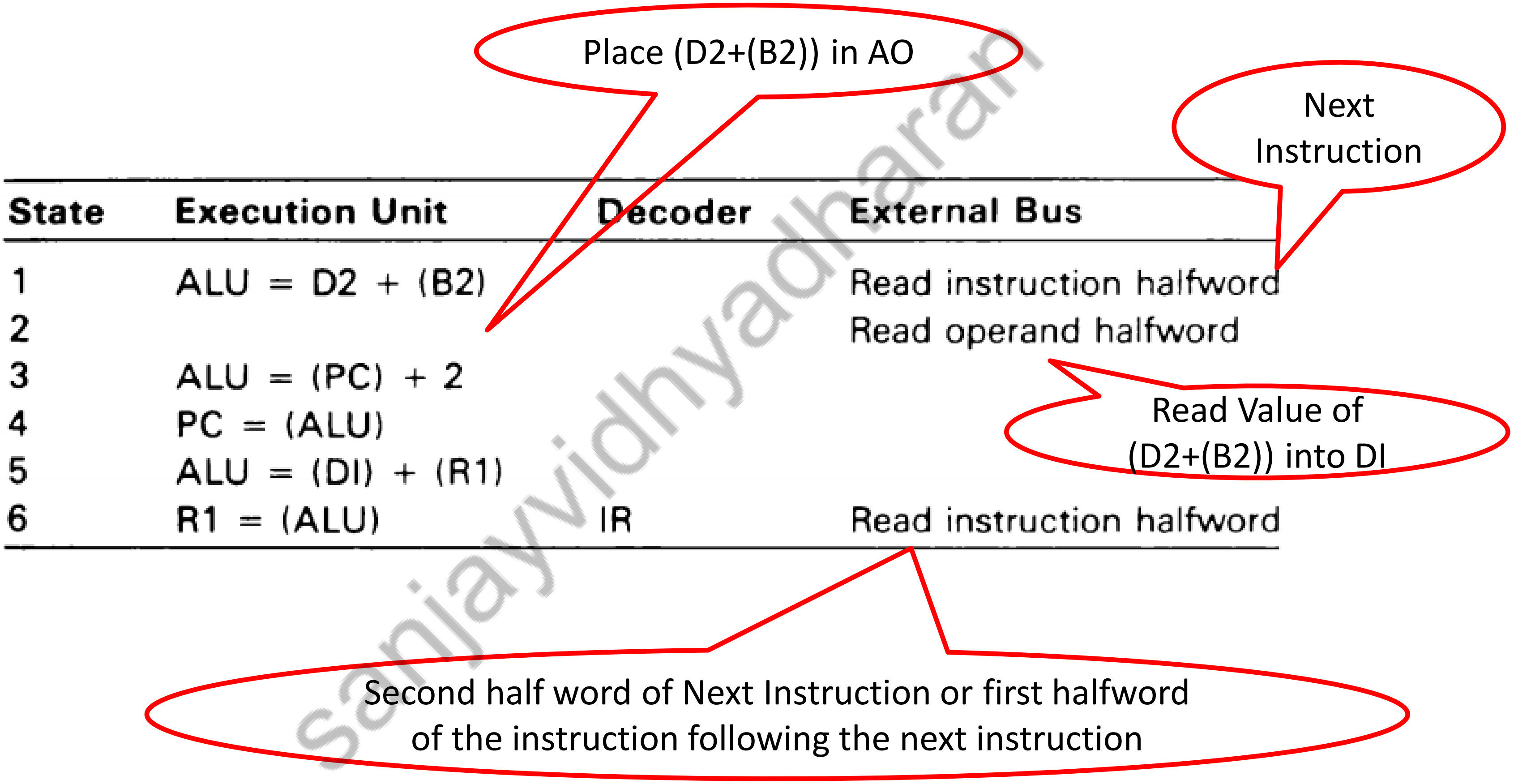


State	Execution Unit	Decoder	External Bus
1			Read instruction halfword
2	ALU = D2 + (B2)		
3			Read operand halfword
4	ALU = (DI) + (R1)		
5	R1 = (ALU)		
6	ALU = (PC) + 2		
7	PC = (ALU)		
8	•		Read instruction halfword
9		IR	

From IRE

ALU data D2+(B2) placed in AO
Read Value of into DI

ADD Instruction Execution with Overlap



State	Execution Unit	Decoder	External Bus
1	ALU = D2 + (B2)		Read instruction halfword
2			Read operand halfword
3	ALU = (PC) + 2		
4	PC = (ALU)		
5	ALU = (DI) + (R1)		
6	R1 = (ALU)	IR	Read instruction halfword

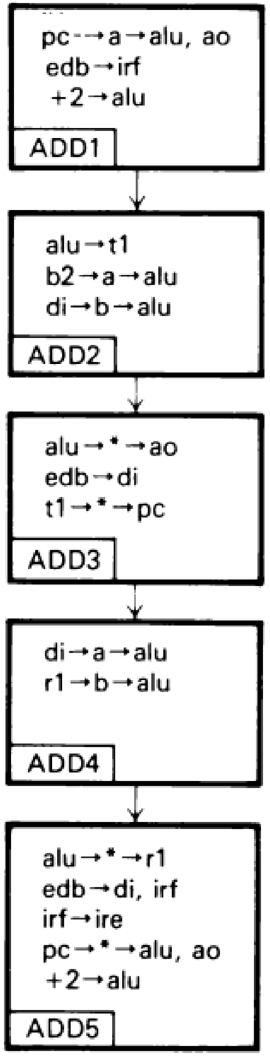
Place (D2+(B2)) in AO

Next Instruction

Read Value of (D2+(B2)) into DI

Second half word of Next Instruction or first halfword of the instruction following the next instruction

ADD Instruction Timing State



pc→a→alu, ao
edb→irf
+2→alu

ADD1

- Increment program counter
- Read next instruction halfword
- The sum is stored in a register in the ALU

Place data into ALU in first Clock

alu→t1
b2→a→alu
di→b→alu

ADD2

- Save the ALU (updated program counter)
- B2 register to internal A bus to ALU
- Displacement D2 from DI to B to ALU
- The sum is stored in a register in the ALU

Transfer data from ALU to T1

alu→*→ao
edb→di
t1→*→pc

ADD3

- Send the operand address to the pads
- Read the operand into DI from the pads
- Save the updated program counter

Read and write from ALU at the same Time

di→a→alu
r1→b→alu

ADD4

- Add the operands

alu→*→r1
edb→di, irf
irf→ire
pc→*→alu, ao
+2→alu

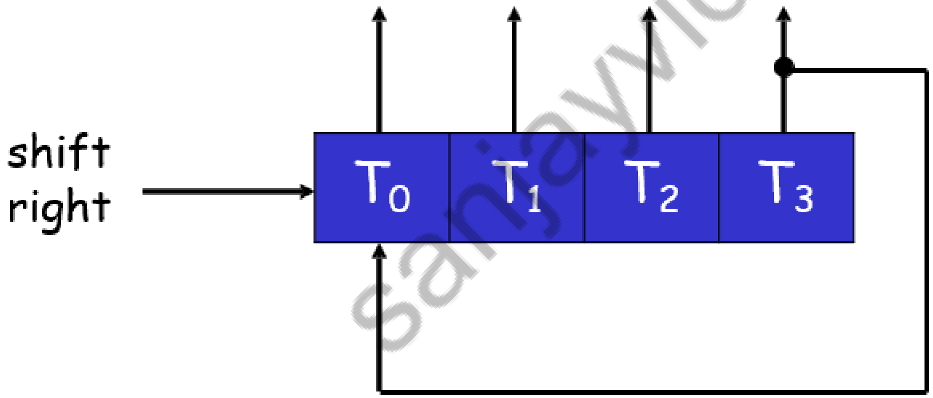
ADD5

- Store the result
- Read second instruction halfword
- Decode next instruction
- Update the PC and read instruction halfword

Read and write from IRF at the same Time

Timing Signals

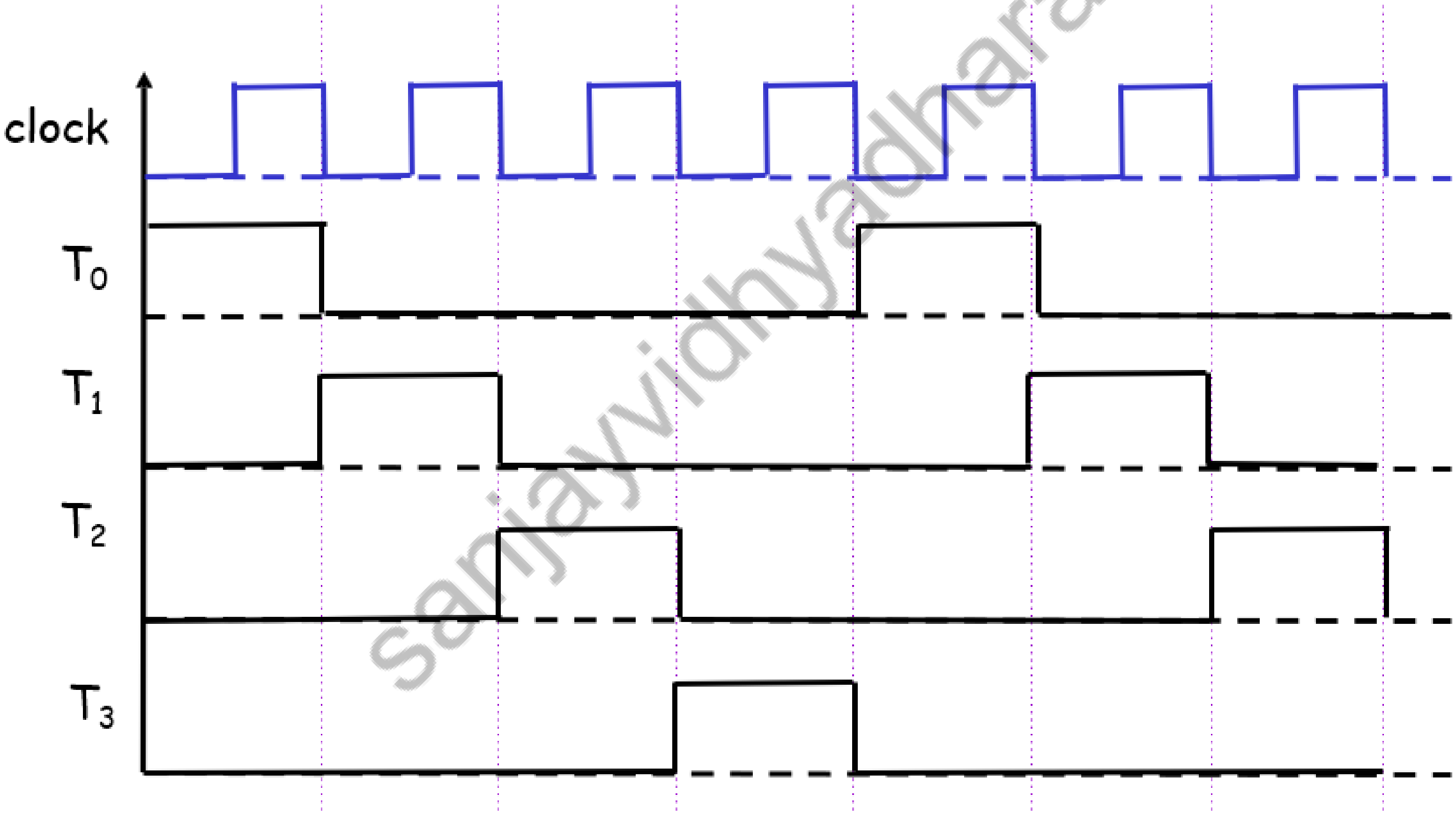
- Ring Counter
  - Timing signals control the sequence of operations in a digital system
  - A ring counter is a circular shift register with only one flip-flop being set at any particular time, all others are cleared.



initial value 1000

Timing Signals

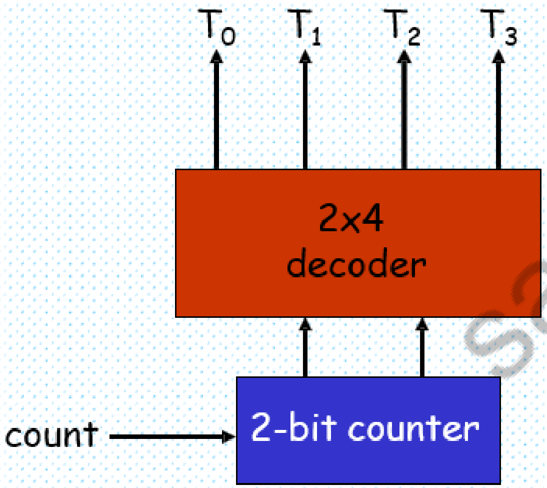
- Sequence of timing signals



Timing Signals

Counter with Decoder

- To generate 2^n timing signals,
  - we need a shift register with 2^n flip-flops
- or, we can construct the ring counter with a binary counter and a decoder



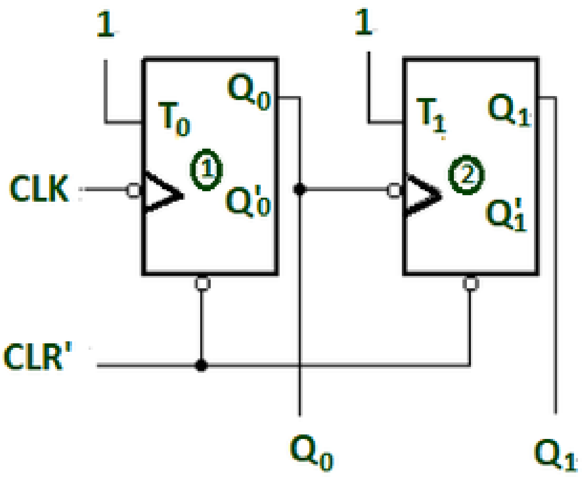
<u>Cost:</u>

•2 flip-flop

•2-to-4 line decoder

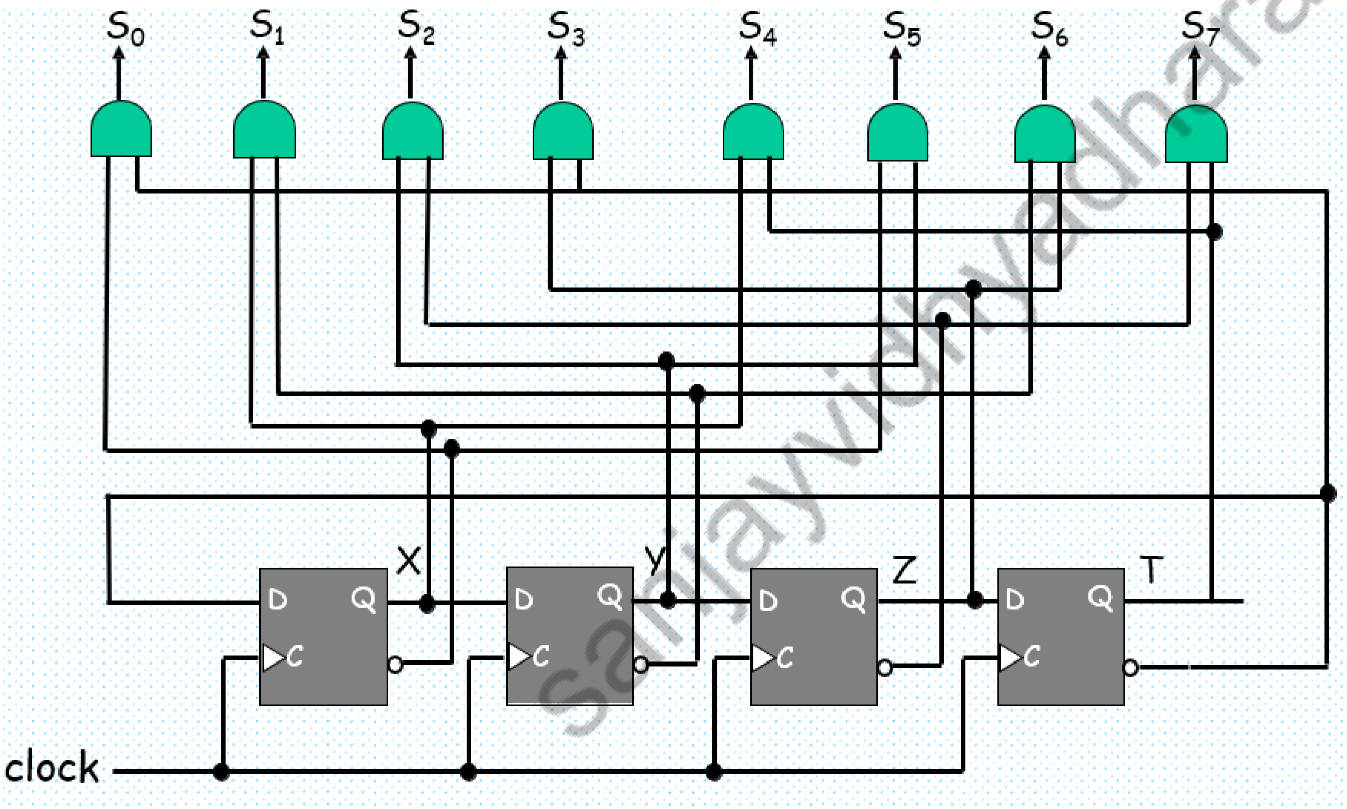
<u>Cost in general case</u>:

- n flip-flops
- n-to-2^n line decoder
  - 2^n n-input AND gates



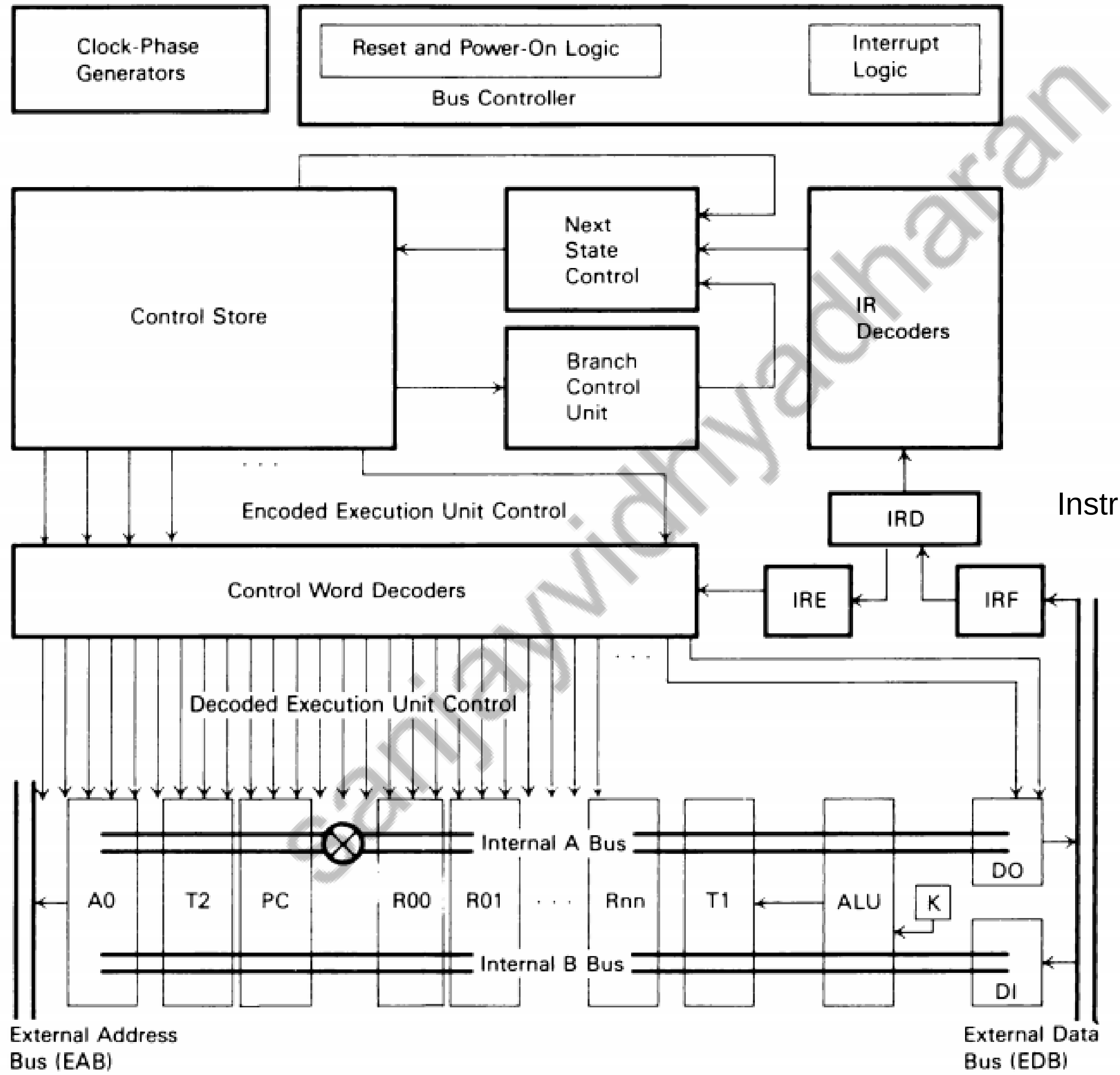
Timing Signals

Johnson Counter



sequence number	Flip-flop outputs			
	X	y	Z	T
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

Timing Signals

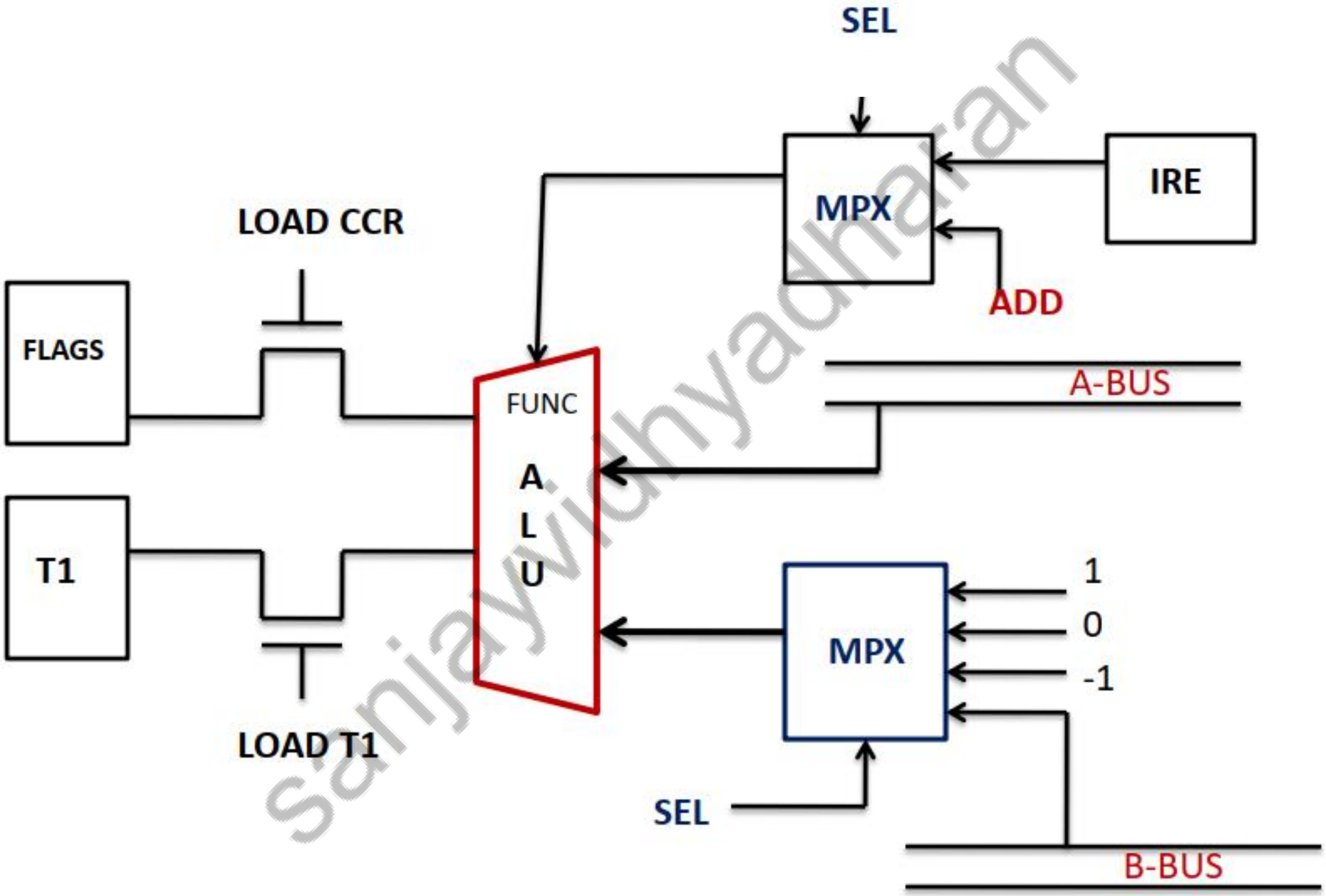


Instruction register for decode (IRD)

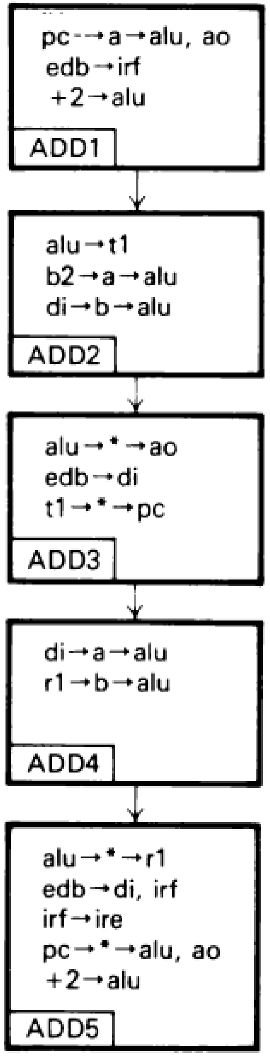
Timing Signals

- During phase 1, the source register is gated to the internal bus.
- During phase 2, the signal on the internal bus is amplified and broadcast the length of the bus.
- During phase 3, the signal on the internal bus is gated to the destination.
- In phase 4, the bus is returned to a neutral state.

Timing Signals



Timing Signals



pc→a→alu, ao
edb→irf
+2→alu
ADD1

- Increment program counter
- Read next instruction halfword
- The sum is stored in a register in the ALU

alu→t1
b2→a→alu
di→b→alu
ADD2

- Save the ALU (updated program counter)
- B2 register to internal A bus to ALU
- Displacement D2 from DI to B to ALU
- The sum is stored in a register in the ALU

alu→*→ao
edb→di
t1→*→pc
ADD3

- Send the operand address to the pads
- Read the operand into DI from the pads
- Save the updated program counter

di→a→alu
r1→b→alu
ADD4

- Add the operands

alu→*→r1
edb→di, irf
irf→ire
pc→*→alu, ao
+2→alu
ADD5

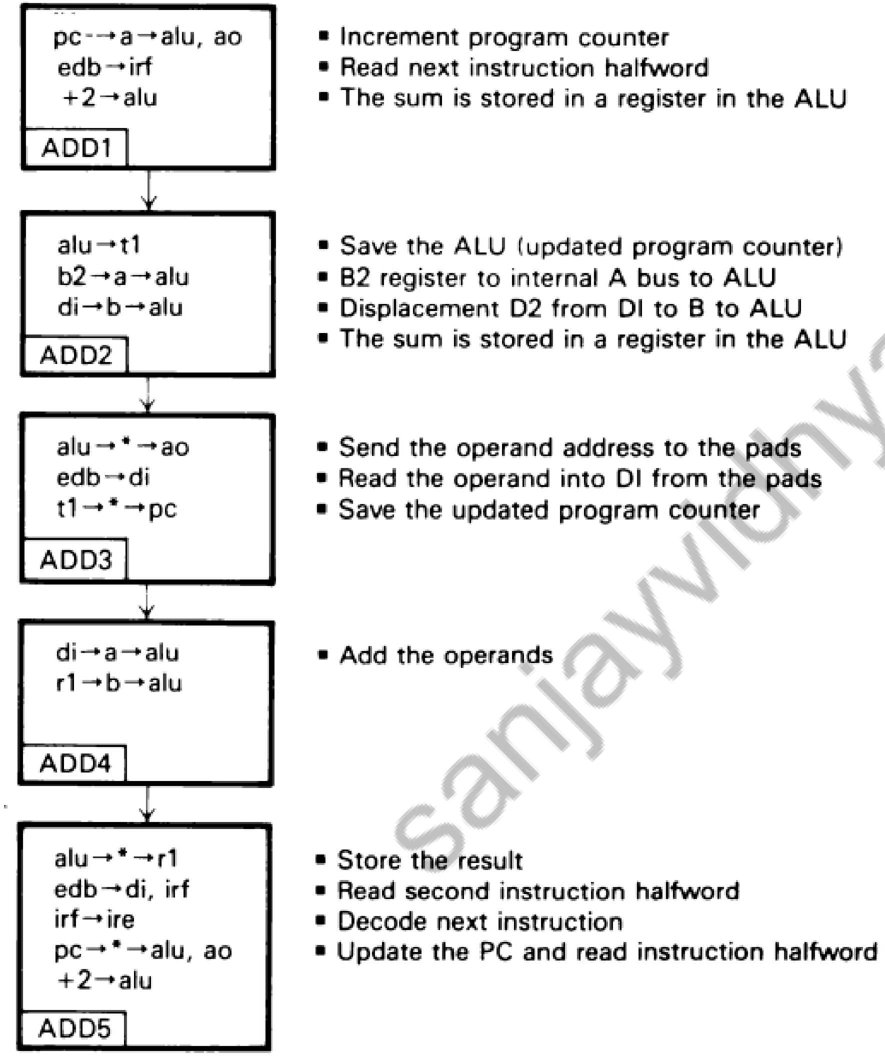
- Store the result
- Read second instruction halfword
- Decode next instruction
- Update the PC and read instruction halfword

In state ADD1,
The PC will reach AO at phase 3.
AO is the buffer that drives the pads.
The instruction returned by the memory on the EDB will arrive at phase 4 or (some) phase 4 prime .

In ADD2,
The value being saved in T1 is the updated PC .
The ALU-to-T1 transfer must be a phase 1 transfer. It will take 3 phases for data from b2 and di to reach ALU

In ADD3,
PC is updated with T1.
T1 not loaded with ALU.

Timing Signals



pc→a→alu, ao
edb→irf
+2→alu

ADD1

- Increment program counter
- Read next instruction halfword
- The sum is stored in a register in the ALU

alu→t1
b2→a→alu
di→b→alu

ADD2

- Save the ALU (updated program counter)
- B2 register to internal A bus to ALU
- Displacement D2 from DI to B to ALU
- The sum is stored in a register in the ALU

alu→*→ao
edb→di
t1→*→pc

ADD3

- Send the operand address to the pads
- Read the operand into DI from the pads
- Save the updated program counter

di→a→alu
r1→b→alu

ADD4

- Add the operands

alu→*→r1
edb→di, irf
irf→ire
pc→*→alu, ao
+2→alu

ADD5

- Store the result
- Read second instruction halfword
- Decode next instruction
- Update the PC and read instruction halfword

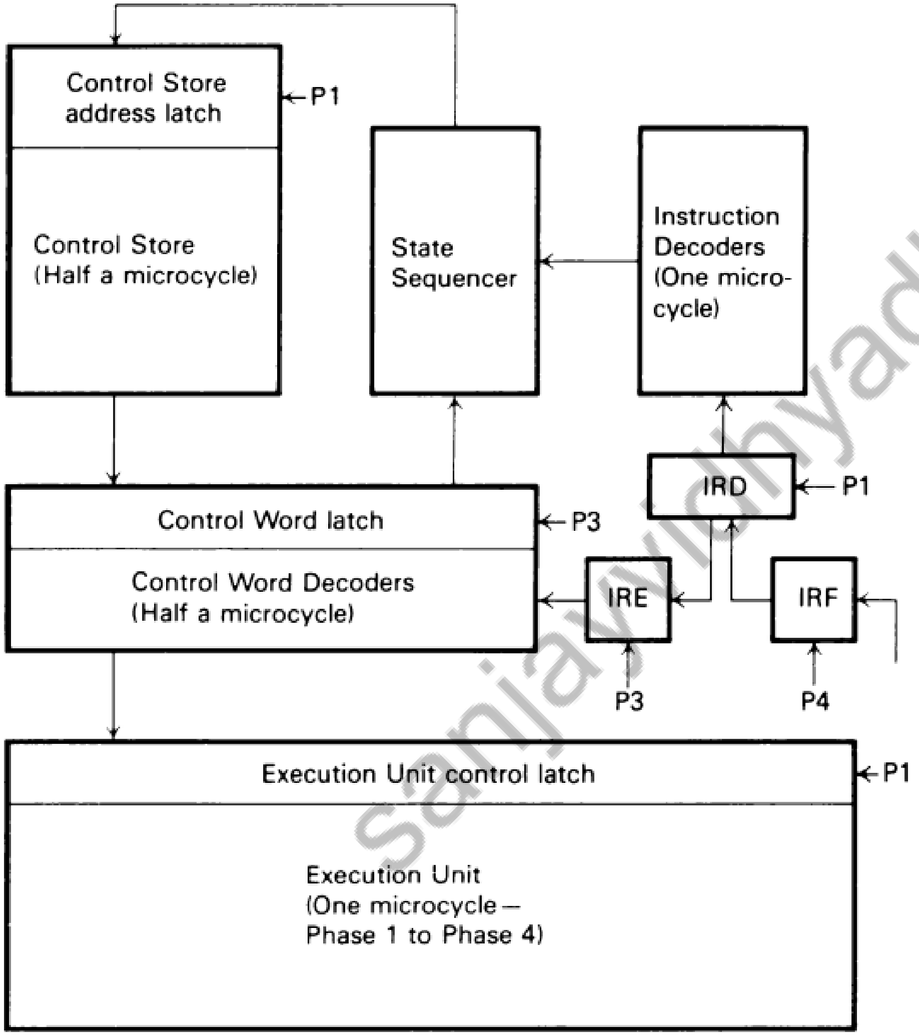
In state ADD4,
The register DI and the register designated by the R1 field of instruction are gated to buses B and A, respectively, during phase 1.

During phase 2, the contents of the registers are amplified and broadcast down the bus.

During phase 3, the inputs to the ALU are opened, and the ALU begins to operate.

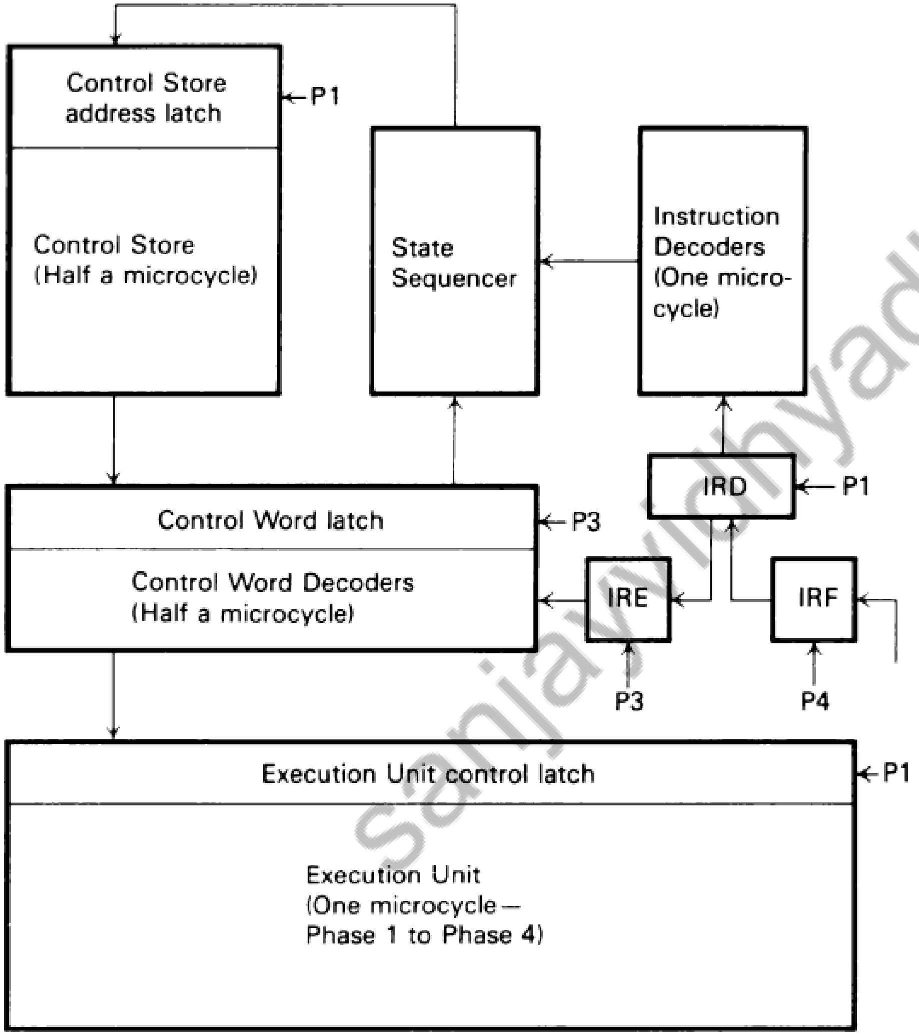
During phase 4, the result of the ALU operation is saved in the ALU output register, and the resulting condition codes are sent to the condition code register.

Timing Signals


Control Store address latch
P1
Control Store (Half a microcycle)
State Sequencer
Instruction Decoders (One micro-cycle)
IRD
P1
Control Word latch
P3
Control Word Decoders (Half a microcycle)
IRE
IRF
P3
P4
Execution Unit control latch
P1
Execution Unit (One microcycle – Phase 1 to Phase 4)

- The execution unit operates on a four-phase clock .
- A microcycle is one sequence of the four clock phases.
- The instruction decoders take one microcycle, and the control store takes half a microcycle. The control word decoders also take half a microcycle.
- Output of the control word decoders must be from the beginning of phase 1 through the end of phase 4
- Control Store and Control word decoder require 1 micro-cyle hence control store address latch should be latched beginning of phase 1.

Timing Signals

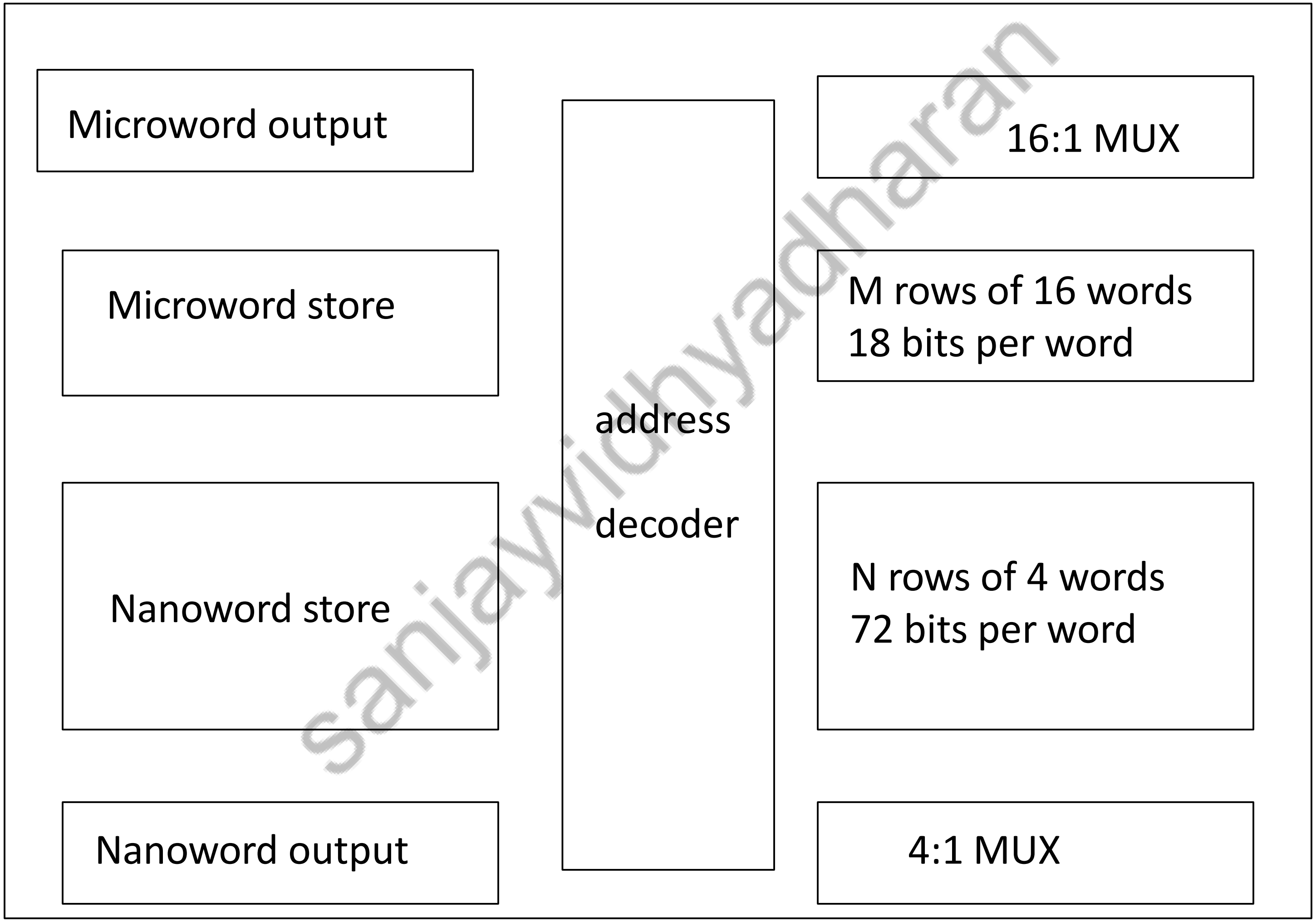


Control Store address latch
P1
State Sequencer
Instruction Decoders (One micro-cycle)
Control Store (Half a microcycle)
IRD
P1
Control Word latch
P3
Control Word Decoders (Half a microcycle)
IRE
IRF
P3
P4
Execution Unit control latch
P1
Execution Unit (One microcycle – Phase 1 to Phase 4)

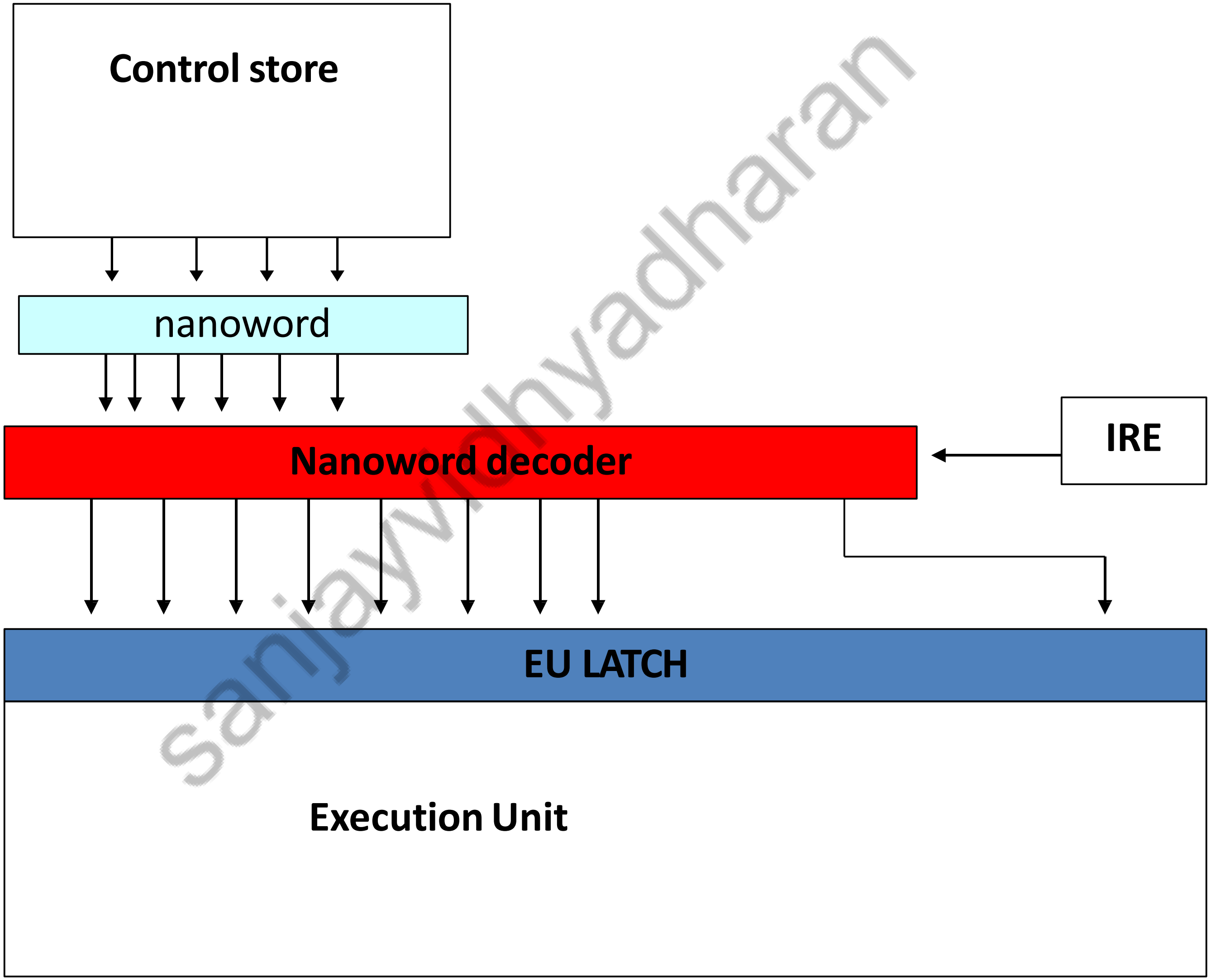
- Since the IR decoders take one microcycle, their input must be available at phase-1 two microcycles before the execution unit is to operate. When we fetch an instruction, it is loaded into the IRF on phase 4. At the next phase-1, we could load the IRE and begin instruction decoding.

Two-level control store (organization)

Motorola MC68000

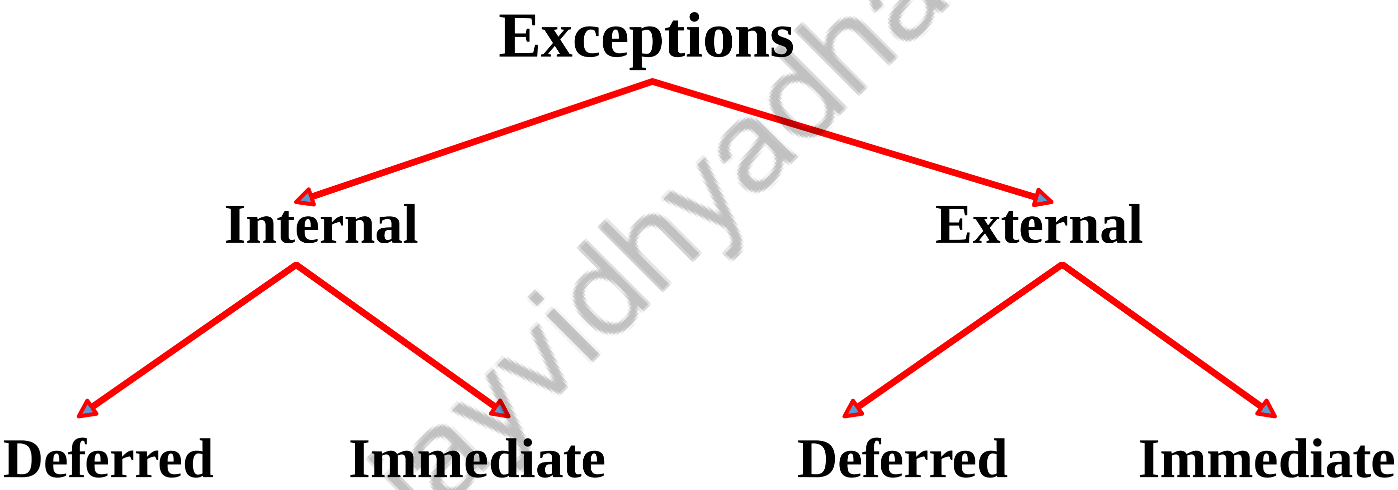


Two-level control store (organization)



Exceptions

Anything that isn’t a valid instruction is an Exception.



Exceptions

Internal

External

Deferred

Immediate

Deferred

Immediate

Exceptions

Interrupts

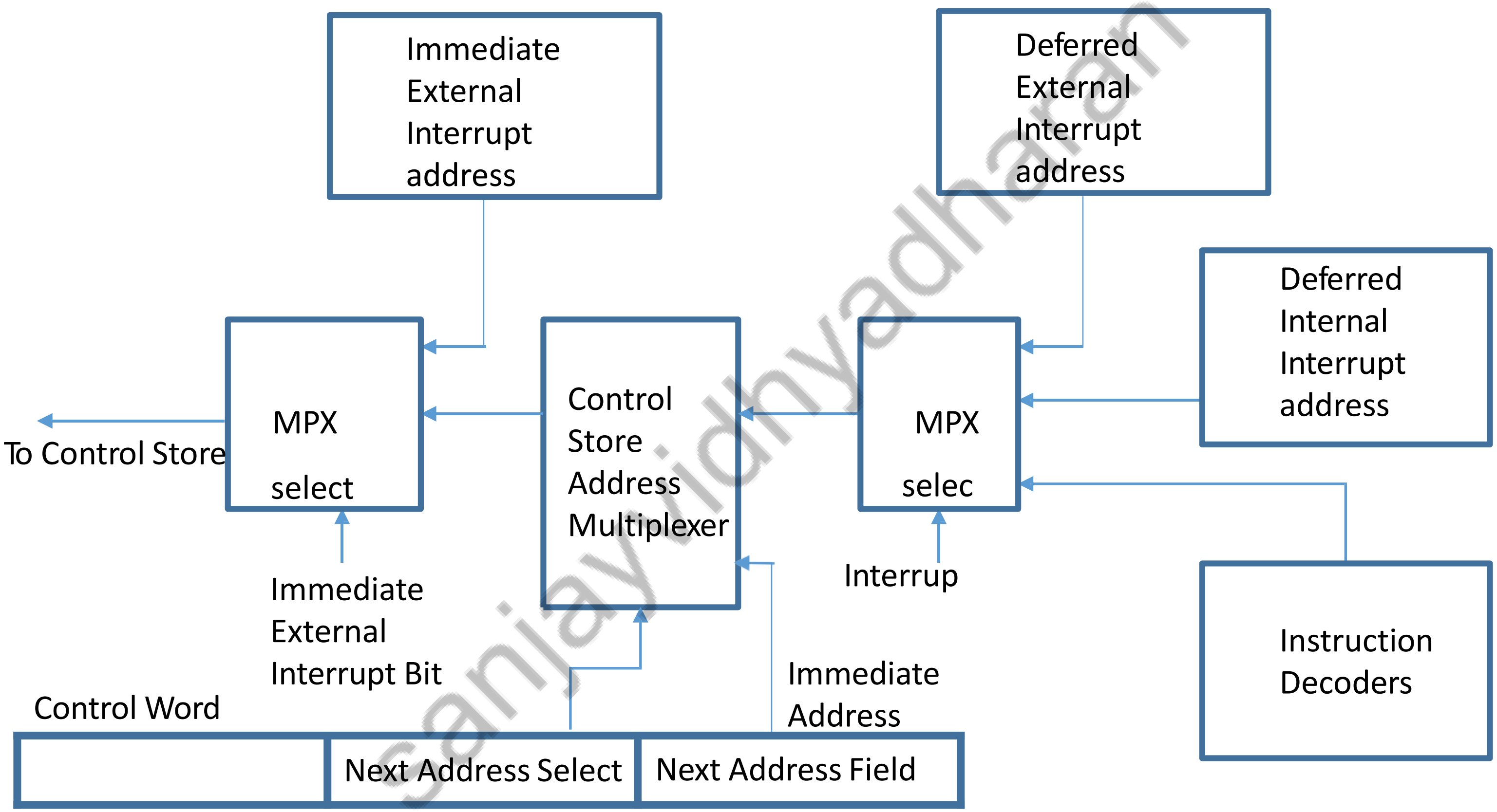
An interrupt is a request for a change in the normal instruction sequence. From the microprocessor's view, there are two kinds of interrupts.

One kind comes from inside the chip (internal interrupts), and the other kind comes from outside the chip (external interrupts) .

The following might cause an internal interrupt:
fixed-point overflow
divide-by-zero
trace

These conditions might cause an external interrupt :
bus error
peripheral (chip) service request
reset request
power-on
test request

Exceptions


Immediate External Interrupt address
Deferred External Interrupt address
Deferred Internal Interrupt address
MPX select
To Control Store
Control Store Address Multiplexer
MPX selec
Interrup
Immediate External Interrupt Bit
Instruction Decoders
Immediate Address
Control Word
Next Address Select
Next Address Field

Thank you