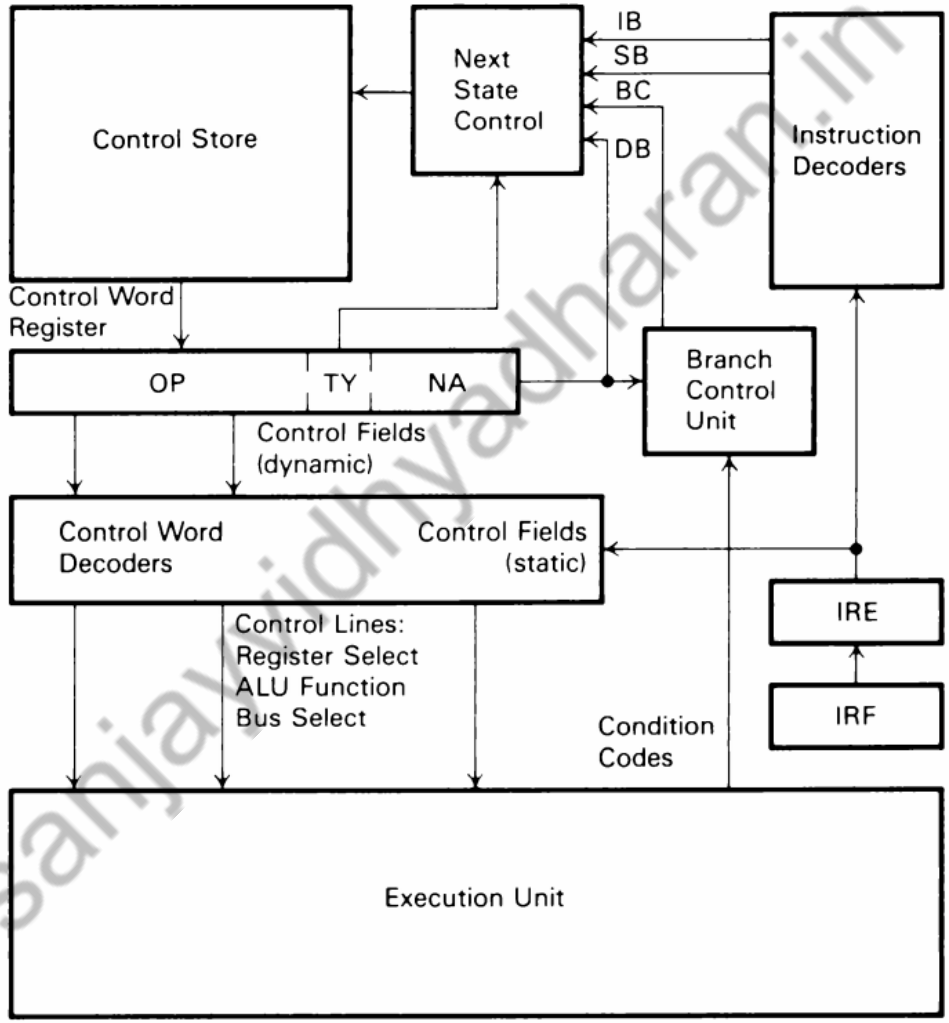
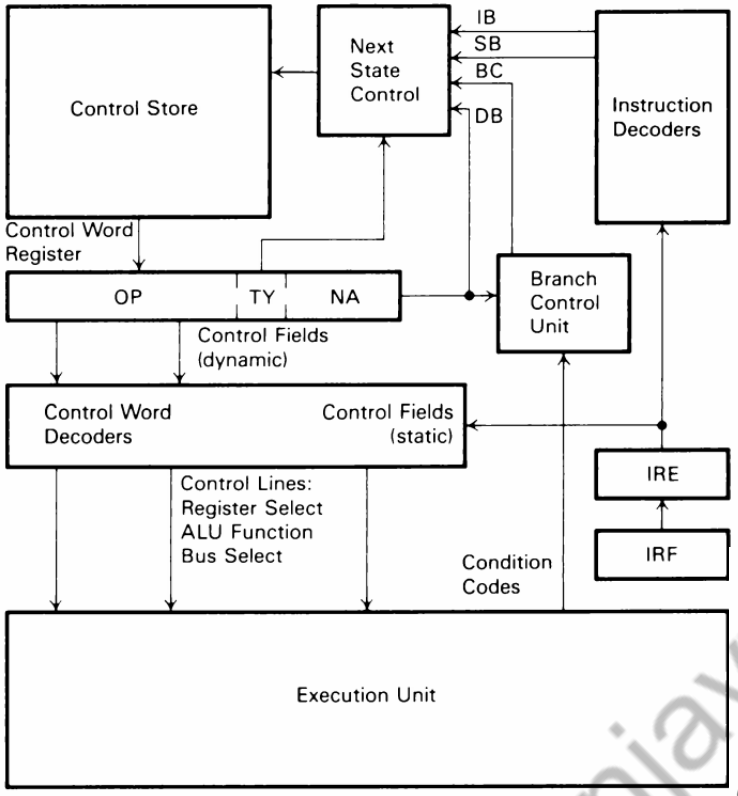
VLSI SYSTEMS AND ARCHITECTURE
2021-22
Lecture 9
Flowcharts to Datapath Control Design

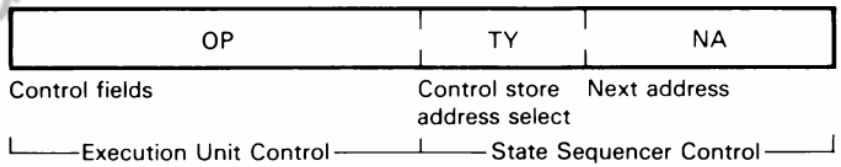
By Dr. Sanjay Vidhyadharan

MIN execution unit block diagram



Control Word Format





OP: Control Fields
Small fields of bits are decoded (by the control word logic decoders) to drive control lines in the execution unit and controller.

TY: Control Store Address Select
Next address (type) select

BC: Branch Conditionally—next control store address is NA modified by a condition code from the execution unit

DB: Direct Branch—next control store address is NA

IB: Instruction Branch—next control store address is from the control word decoders using IRE (for the next instruction)

SB: Sequence Branch—next control store address is from the control word decoders using IRE (for the next sequence to help execute the current instruction)

NA: Next Address
Next state (direct) address

Execution Sequences for Register-to-Register

PC Control

LOAD

ry→a→alu, rx 0→alu	edb→irf pc→a→alu, ao +1→alu
	irf→ire t1→a→pc

STORE

rx→a→alu, ry 0→alu	edb→irf pc→a→alu, ao +1→alu
	irf→ire t1→a→pc

ADD

rx→a→alu ry→b→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→ry	irf→ire t1→a→pc

SUB

rx→a→alu ry→b→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→ry	irf→ire t1→a→pc

Execution Sequences with a Memory Operand Reference

PC Control

ADD

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

AND

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

SUB

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

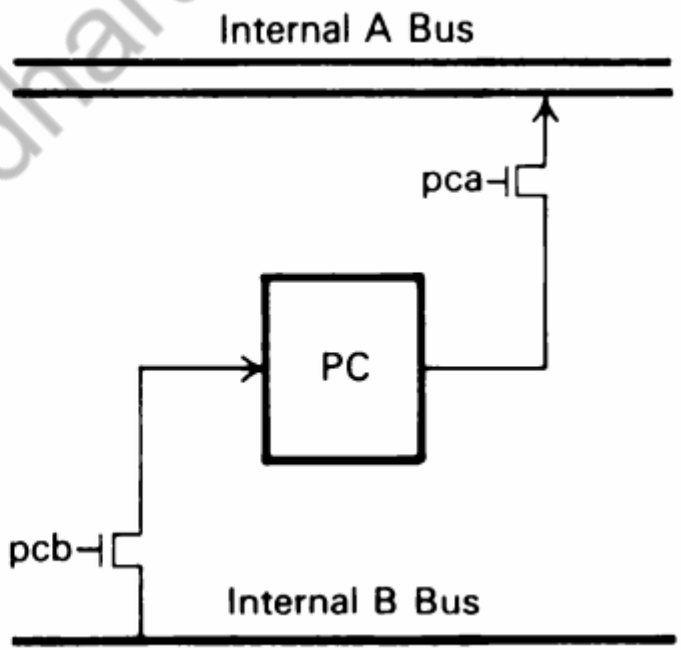
OP CODE

PC Control

pc→a
a→pc
b→pc
none

WE could just as easily write to PC using the B bus, so I can eliminate a-> pc, thus simplifying the control log

pcb \ pca	0	1
0	none	pc→a
1	b→pc	x



Control Fields

AO	PC	T2	Regs	T1	ALU	K	DI	DO	IRE	IRF	

OP CODE

T2 Control

Execution Sequences with a Memory Operand Reference

ADD

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

AND

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

SUB

di→b→alu rx→a→alu	edb→irf pc→a→alu, ao +1→alu
t1→a→do t2→b→ao	irf→ire t1→b→pc

OP CODE

T2 Control

t2→a

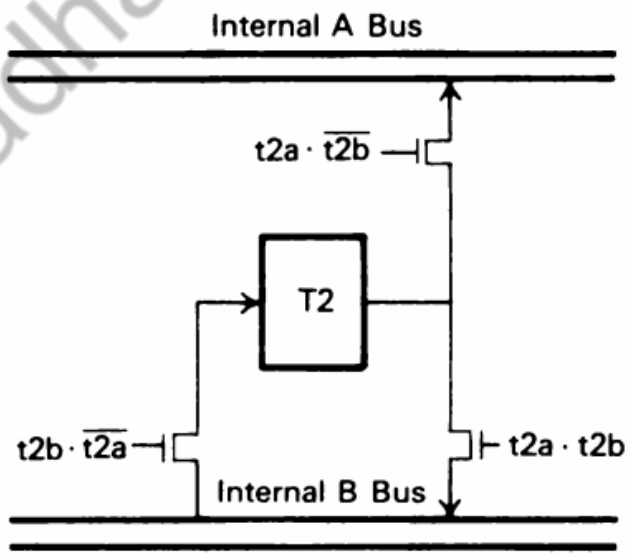
t2→b

a→t2 (only one occurrence of this one)

b→t2

none

t2b \ t2a	0	1
0	none	t2→a
1	b→t2	t2→b

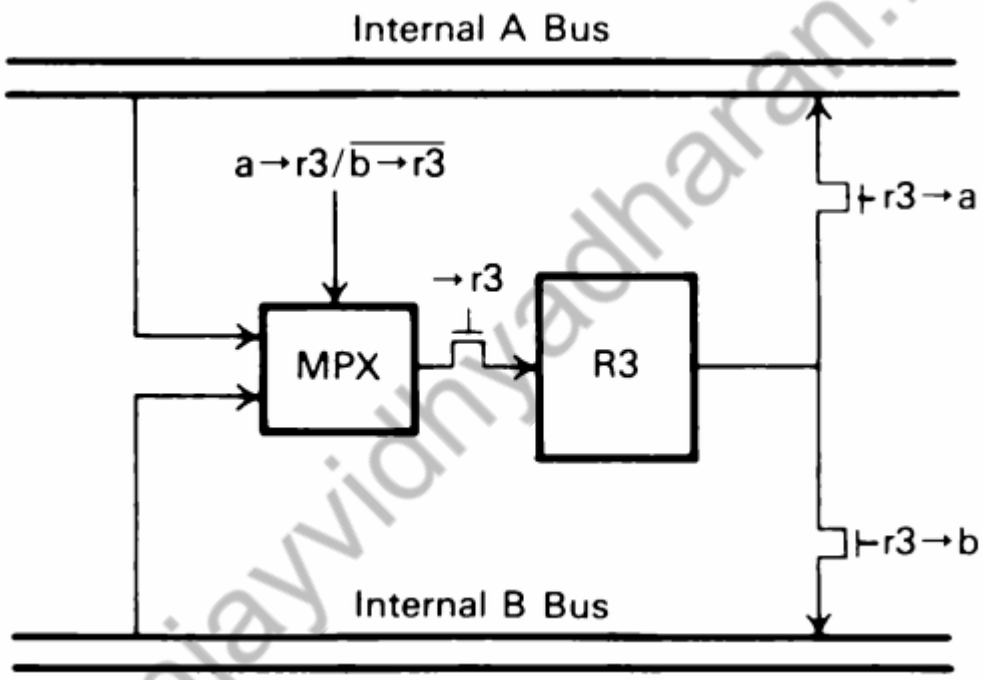


Control Fields

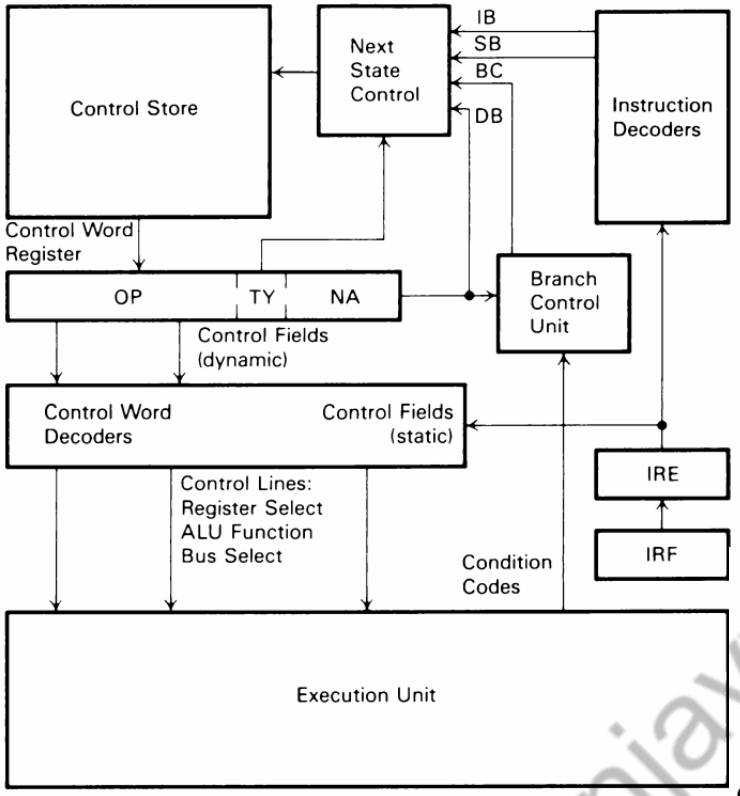
AO	PC	T2	Regs	T1	ALU	K	DI	DO	IRE	IRF

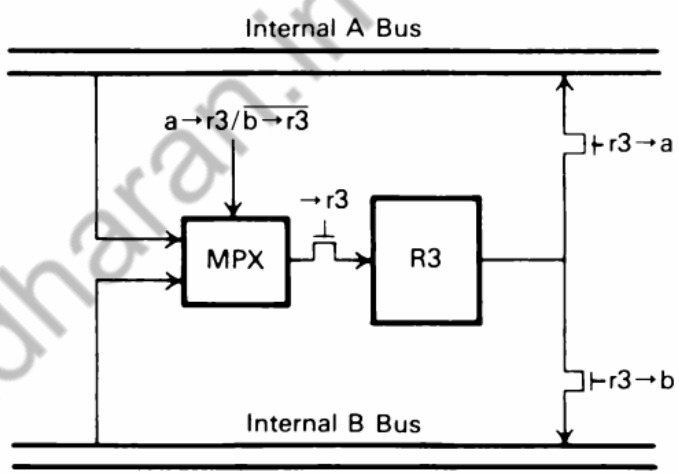
OP CODE

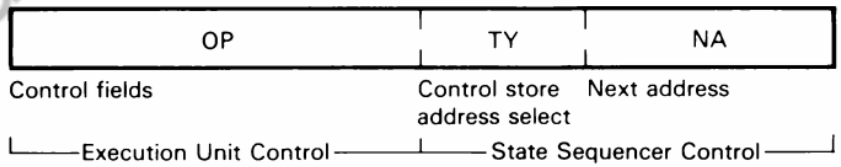
Register Control



Control Word Format







OP: Control Fields
Small fields of bits are decoded (by the control word logic decoders) to drive control lines in the execution unit and controller.

TY: Control Store Address Select
Next address (type) select

BC: Branch Conditionally—next control store address is NA modified by a condition code from the execution unit

DB: Direct Branch—next control store address is NA

IB: Instruction Branch—next control store address is from the control word decoders using IRE (for the next instruction)

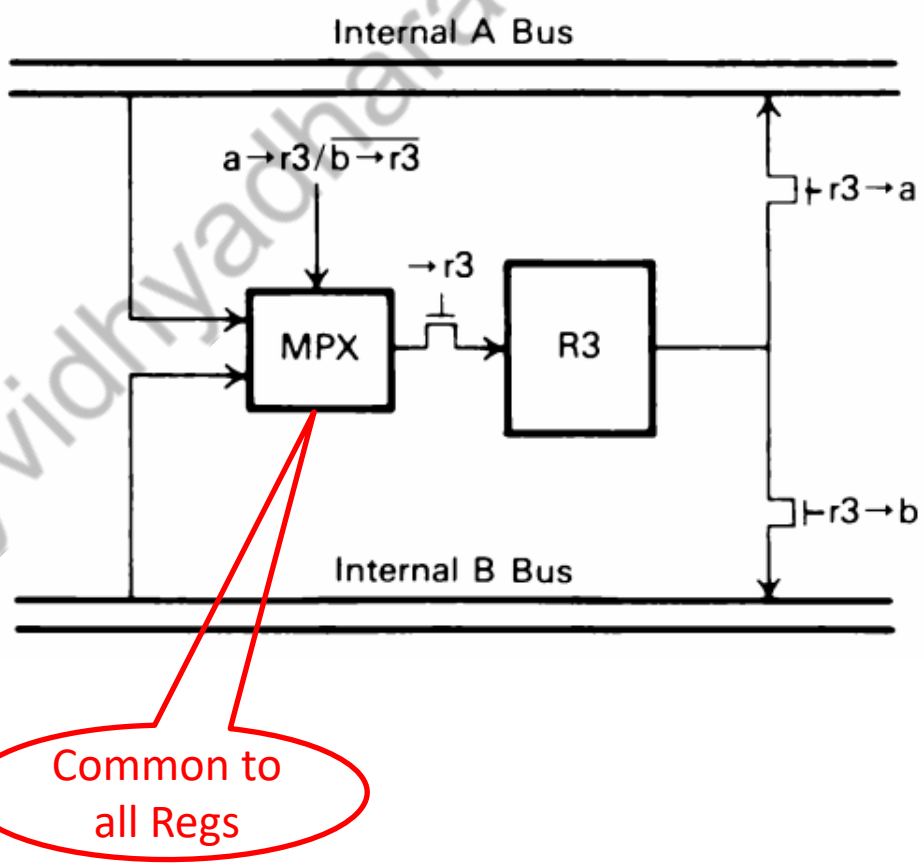
SB: Sequence Branch—next control store address is from the control word decoders using IRE (for the next sequence to help execute the current instruction)

NA: Next Address
Next state (direct) address

OP CODE

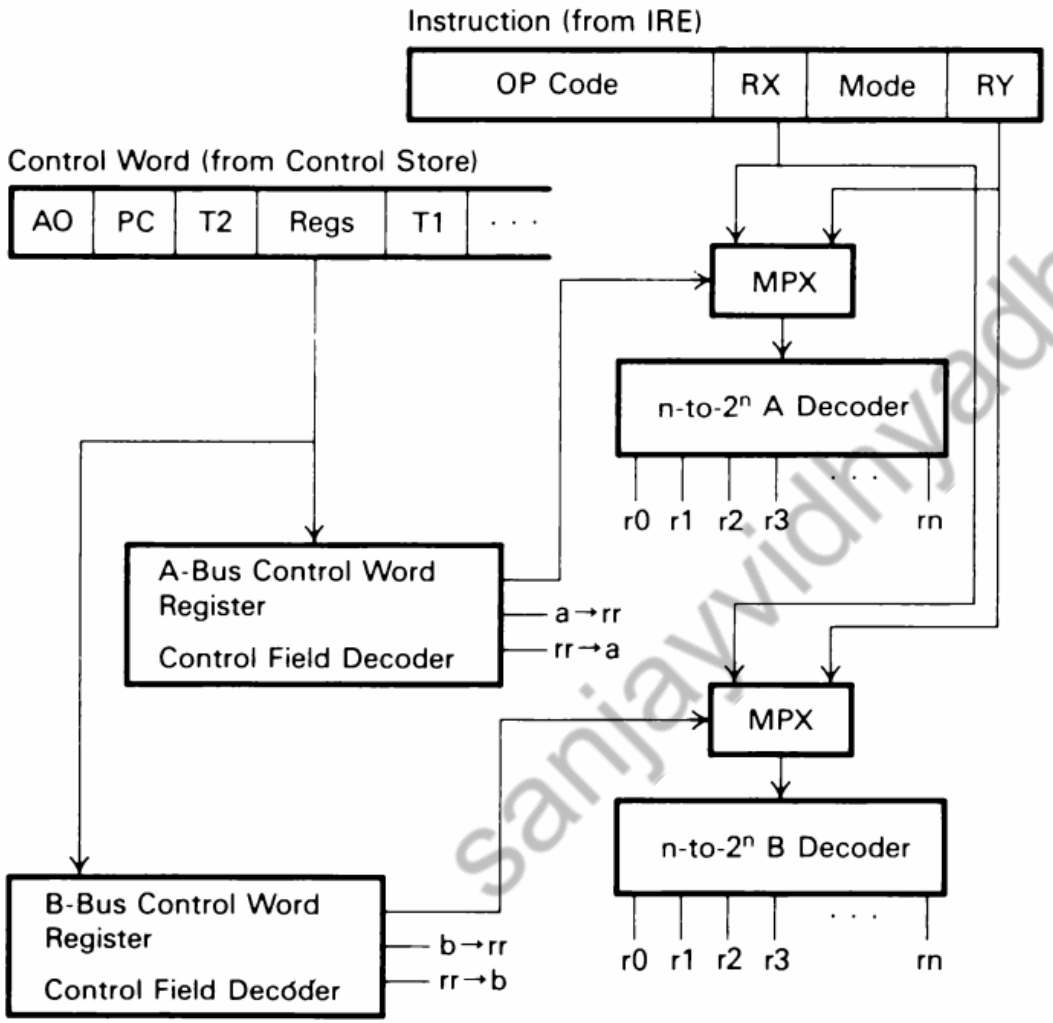
Register Control

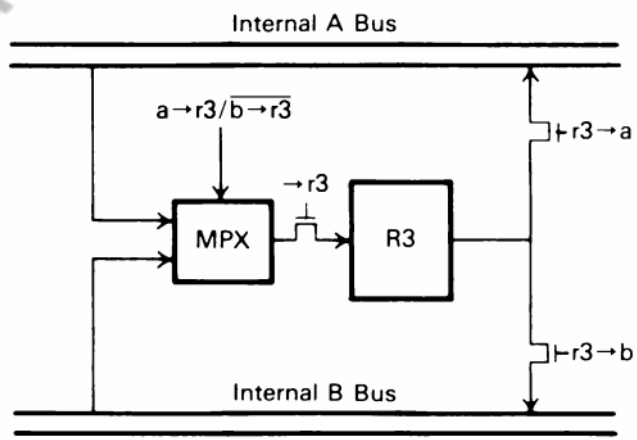
Control Word States (What I see in the flowcharts)
ry→a
b→rx
ry→b;b→rx
rx→a
rx→b;b→ry
rx→a;ry→b
b→ry
b→rx;a→ry
rx→a;b→ry
none



OP CODE

Register Control





OP CODE

Register Control

Control Word States (What I see in the flowcharts)
ry→a
b→rx
ry→b;b→rx
rx→a
rx→b;b→ry
rx→a;ry→b
b→ry
b→rx;a→ry
rx→a;b→ry
none

	00	01	11	10	
00	0 none	1 b→rx	3 rx→a	2 ry→a	
01	4 b→ry	5	7 rx→a b→ry	6 rx→b b→ry	b→ry →ry
11	C	D b→rx a→ry	F	E	a→ry
10	8	9 b→rx ry→b	B rx→a ry→b	A	ry→b
		b→rx →rx	rx→a		

OP CODE

Register Control

	00	01	11	10
00	0 none	1 b→rx	3 rx→a	2 ry→a
01	4 b→ry	5	7 rx→a b→ry	6 rx→b b→ry
11	C	D b→rx a→ry	F	E
10	8	9 b→rx ry→b	B rx→a ry→b	A
		b→rx →rx	rx→a	

b→ry

→ry

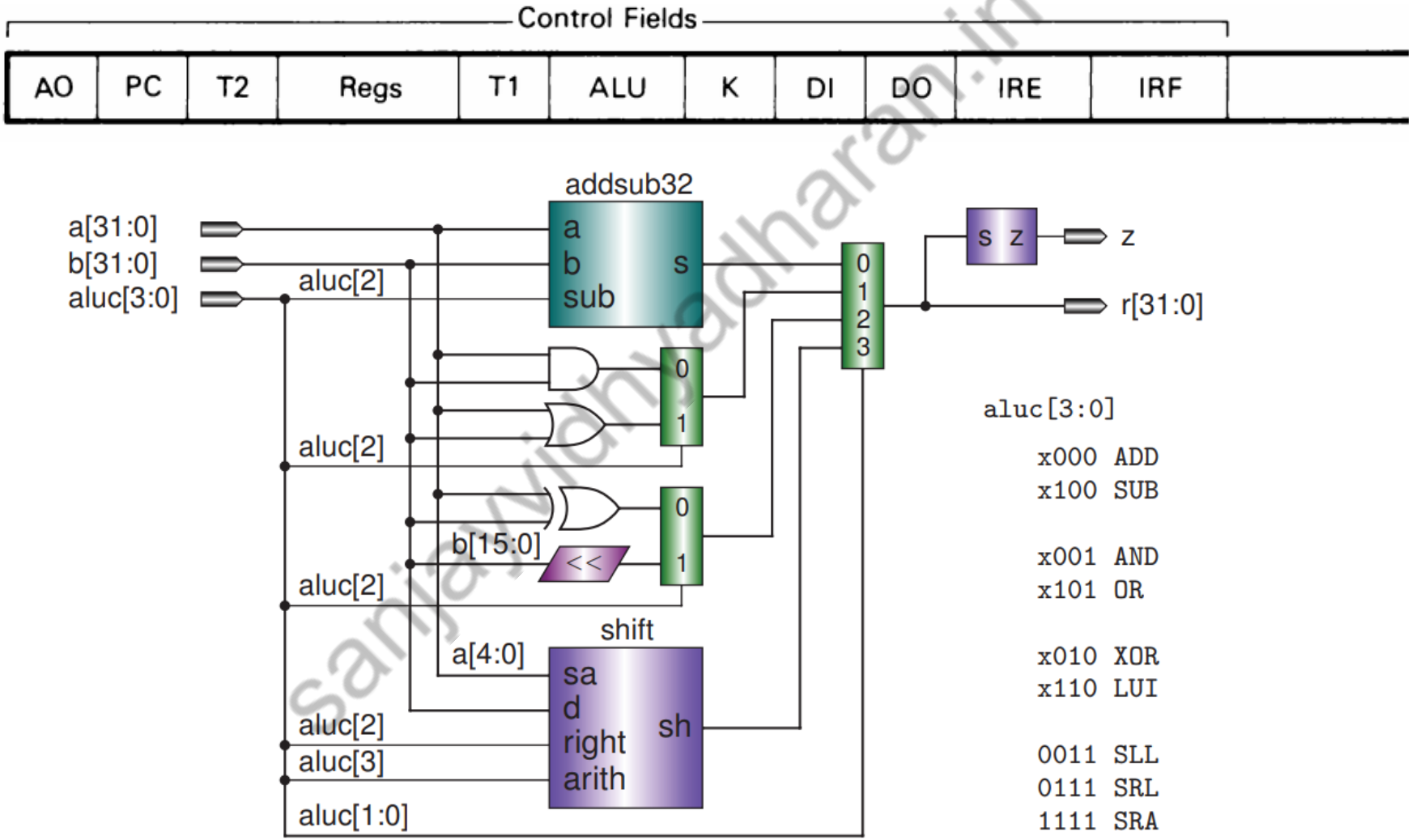
a→ry

ry→b

Control Lines	Decoder Patterns (Control line is active any time the control field matches decoder pattern)
rx→a	xx11
ry→a	0010
rx→b	0110
ry→b	10xx
→rx	xx01
→ry	x1xx
a→ry	11xx
b→rx	xx01
b→ry	01xx

OP CODE

ALU Control



OP CODE

ALU Control

- All instructions use ALU for ADD to update PC
- ALU used for operand address calculation
- Arithmetic and logical instructions use ALU for the requisite operation

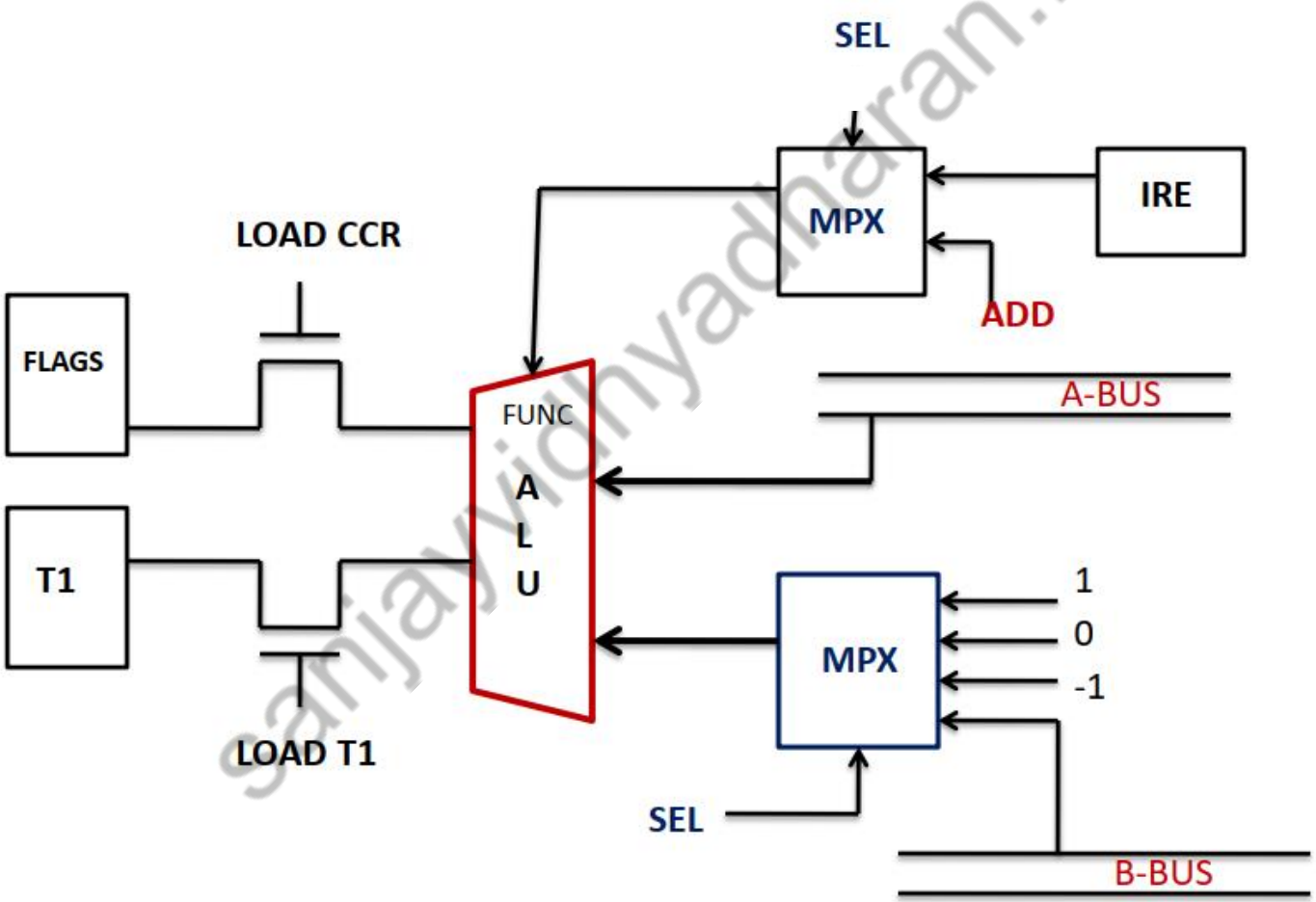
OP CODE

ALU Control

a → alu ;	+1 → alu ;	add-n ;	alu → t1	PC
a → alu ;	b → alu ;	add-n ;	alu → t1	Ry + d
a → alu ;	0 → alu ;	add-s ;	alu → t1	Check Sign, Zero etc
a → alu ;	b → alu ;	op-s ;	alu → t1	ADD/AND etc
a → alu ;	-1 → alu ;	add-n ;	alu → t1	PUSH

OP CODE

ALU Control



OP CODE

ALU Control

load t1

load ccr

add / op select

alu b input select (2 bits)

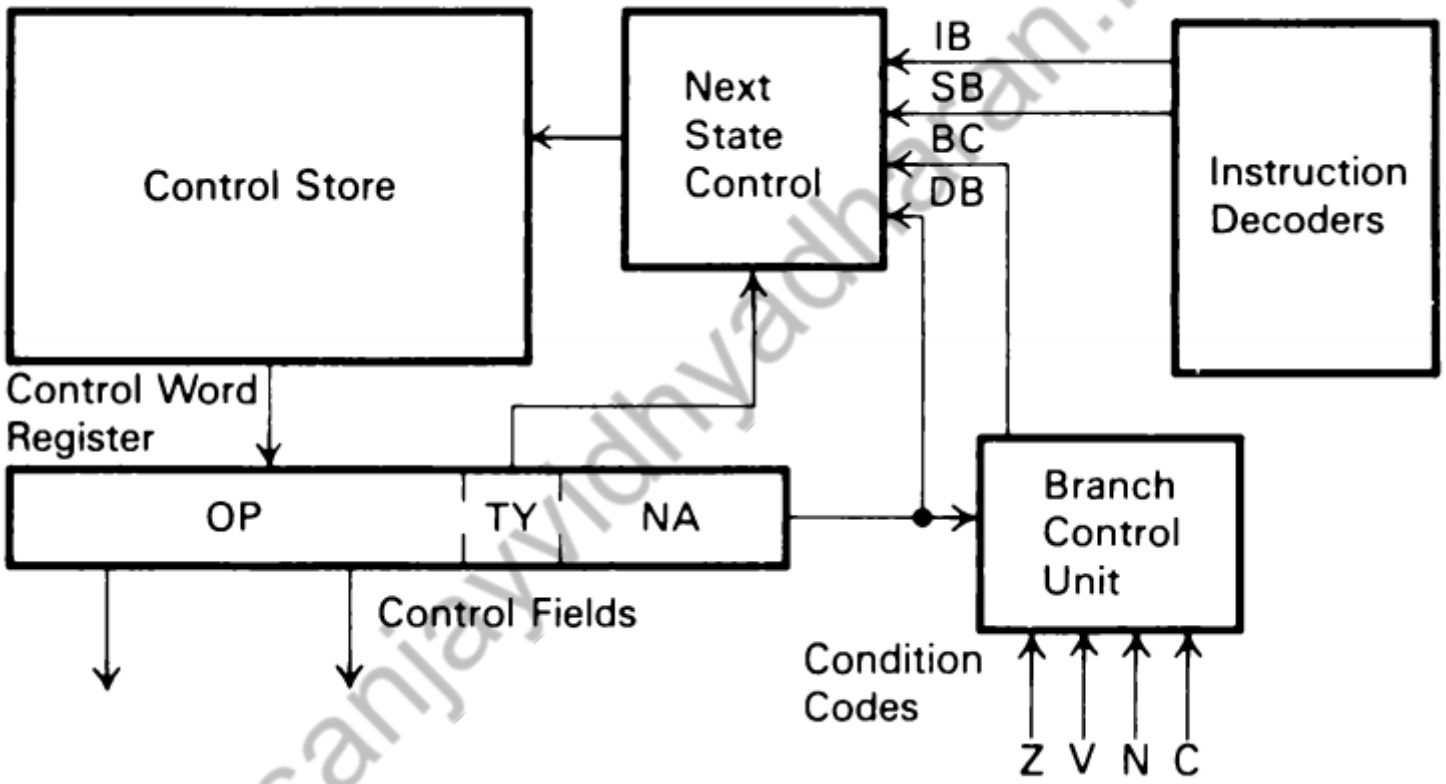
OP CODE

ALU Control

Control word states	decoder pattern
a → alu ; +1 → alu ; add-n ; alu → t1	101
a → alu ; b → alu ; add-n ; alu → t1	010
a → alu ; 0 → alu ; add-s ; alu → t1	100
a → alu ; b → alu ; op-s ; alu → t1	110
a → alu ; -1 → alu ; add-n ; alu → t1	111
none	000

load t1	xxx i000
load ccr	1x0
add / op select	110
alu b input select (2 bits)	x10

State Sequencer



Thank you