



VLSI SYSTEMS AND ARCHITECTURE

2021-22

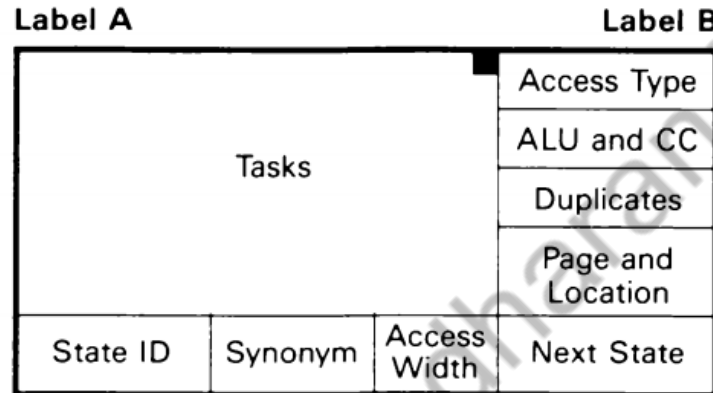
Lecture 8

Hardware Flowcharts Part-3

By Dr. Sanjay Vidhyadharan

Hardware Flowchart

State Identifiers



Sequence Label A

Operation format or instruction type

Example: **rx op ry** → ry
rx op mem → mem

Sequence Label B

Operation sequence or address mode sequence

Example: **ADD, AND, SUB**
Register Indirect Indexed

Access Type

DR Data Read
 DW Data Write
 IR Instruction Read
 NA No Access

ALU and CC (ALU function and Condition Code setting)

Example: **ADD, AND, or OP**
 condition codes
 S Set
 N Not set
 X Don't care

Duplicates

(Not used in MIN)

Page and Location

(Not used in MIN)

Next State

BC Branch Conditionally
 IB Instruction Branch
 SB Sequence Branch
 State ID Direct transfer

Access Width

(Not used in MIN)

Synonym

(Not used in MIN)

State ID

State Identification

■ indicates external bus activity

Hardware Flowchart

Example 1: ADD

RX AR

RY

R → R

ADD

rx → a → alu ry → b → alu			na
			op-s
oprr1			oprr2
edb → irf pc → a → alu, ao t1 → b → ry +1 → alu			ir
			add-n
oprr2			oprr3
irf → ire t1 → b → pc			na
			x-n
oprr3			ib

Label A

Label B

Tasks			Access Type
			ALU and CC
			Duplicates
			Page and Location
State ID	Synonym	Access Width	Next State

Hardware Flowchart

RY → RX

LOAD

edb → irf pc → a → alu, ao ry → b → rx, t2 + 1 → alu			ir
			add-x
ldrr1			ldrr2
irf → ire t1 → b → pc t2 → a → alu 0 → alu			na
			add-s
ldrr2			ib

RX → RY

STORE

edb → irf pc → a → alu, ao rx → b → ry, t2 + 1 → alu			ir
			add-x
strr1			strr2
irf → ire t1 → b → pc t2 → a → alu 0 → alu			na
			add-s
strr2			ib

Hardware Flowchart

Execution Sequences with a Memory Operand Reference

MEM → RX			LOAD
di → b → rx, t2 edb → irf pc → a → alu, ao + 1 → alu			ir
			add-x
ldrm1			ldrm2
irf → ire t1 → b → pc t2 → a → alu 0 → alu			na
			add-s
ldrm2			ib

Label A			Label B
Tasks			Access Type
			ALU and CC
			Duplicates
			Page and Location
State ID	Synonym	Access Width	Next State

Hardware Flowchart

Execution Sequences with a Memory Operand Reference

RX → MEM			STORE		
rx → a → alu, do t2 → b → ao 0 → alu			dw		
			add-s		
strm1			strm2		
edb → irf pc → a → alu, ao + 1 → alu			ir		
			add-n		
strm2			strm3		
irf → ire t1 → b → pc			na		
			x-n		
strm3			ib		

Label A			Label B	
Tasks			Access Type	
			ALU and CC	
			Duplicates	
			Page and Location	
State ID	Synonym	Access Width	Next State	

Hardware Flowchart

Execution Sequences with a Memory Operand Reference

RX OP MEM ADD, AND,
→ MEM SUB

di → b → alu rx → a → alu			na
			op-s
oprm1			oprm2
t1 → a → do t2 → b → ao			dw
			x-n
oprm2			oprm3
edb → irf pc → a → alu, ao + 1 → alu			ir
			add-n
oprm3			oprm4
irf → ire t1 → b → pc			na
			x-n
oprm4			ib

Label A

Label B

Tasks			Access Type
			ALU and CC
			Duplicates
			Page and Location
State ID	Synonym	Access Width	Next State

Hardware Flowchart

(RY + d)@

edb → di pc → a → alu, ao + 1 → alu			ir
			add-n
abdm1			abdm2
t1 → a → pc			na
			x-n
abdm2			abdm3
di → b → alu ry → a → alu			na
			add-n
abdm3			abdm4
edb → di t1 → a → ao, t2			dr
			x-n
abdm4			sb

Label A

Tasks			Access Type
			ALU and CC
			Duplicates
			Page and Location
State ID	Synonym	Access Width	Next State

Label B

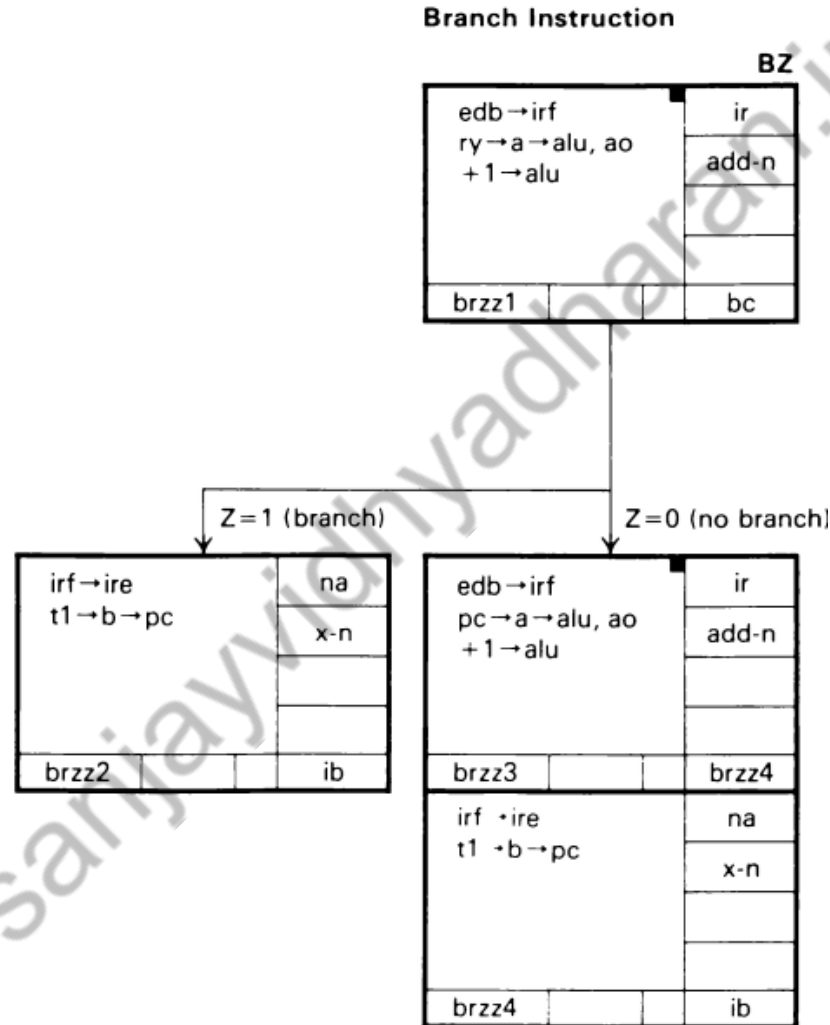
Next State

BC Branch Conditionally
 IB Instruction Branch
 SB Sequence Branch
 State ID Direct transfer

Hardware Flowchart

Z : Zero Flag

JZ (RY)



Hardware Flowchart

**RX OP MEM
→ MEM** **ADD, AND,
SUB**

di → b → alu rx → a → alu		na
		op-s
oprm1		oprm2
t1 → a → do t2 → b → ao		dw
		x-n
oprm2		oprm3
edb → irf pc → a → alu, ao + 1 → alu		ir
		add-n
oprm3		oprm4
irf → ire t1 → b → pc		na
		x-n
oprm4		ib

Branch Instruction

BZ

edb → irf ry → a → alu, ao + 1 → alu		ir
		add-n
brzz1		bc

Z = 1 (branch)

Z = 0 (no branch)

irf → ire t1 → b → pc		na
		x-n
brzz2		ib

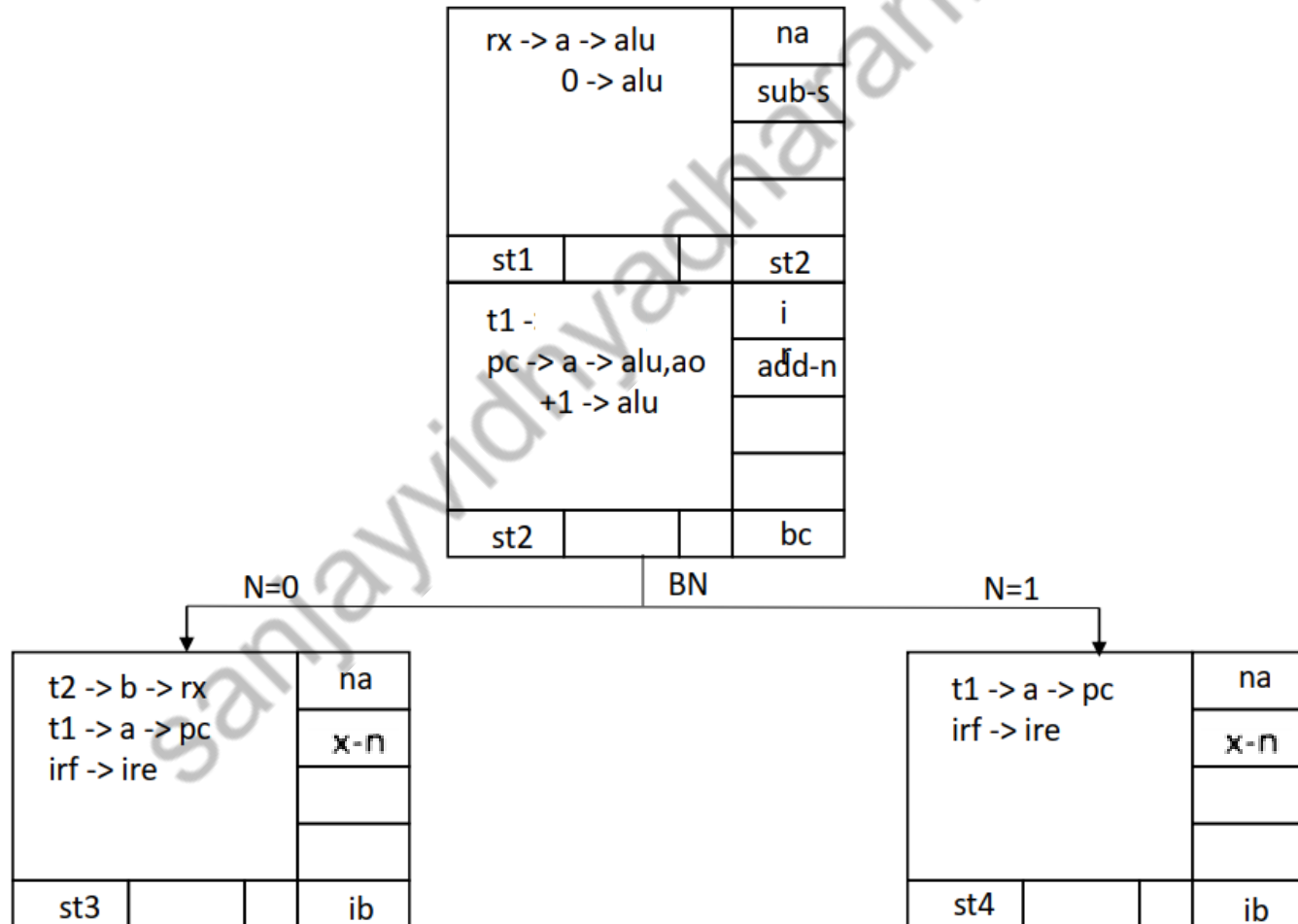
edb → irf pc → a → alu, ao + 1 → alu		ir
		add-n
brzz3		brzz4
irf → ire t1 → b → pc		na
		x-n
brzz4		ib

5/14/2022

10

Hardware Flowchart

MAG RX: This instruction examines the number stored in register RX and stores back the magnitude value of the number in register RX



Hardware Flowchart

SORT @RX @RY :

This instruction compares the two numbers stored in memory at addresses provided by registers RX and RY. It swaps the numbers in the two memory locations if the number at the memory address provided by RY is smaller than the other number.

SORT @RX @RY :



Thank you