CISC Design

Implementation

Virendra Singh

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

EE-309: Microprocessors

Lecture 26 (24 Sep 2015)

CADSL

Control Word Format

Control words

- Operation section (OP) is composed of the fields for Datapath control
- Next state section, containing TY and NA, contains the field for state sequencer control
- If two macro in the Datapath are never used at the same time, you might consider sharing the control field

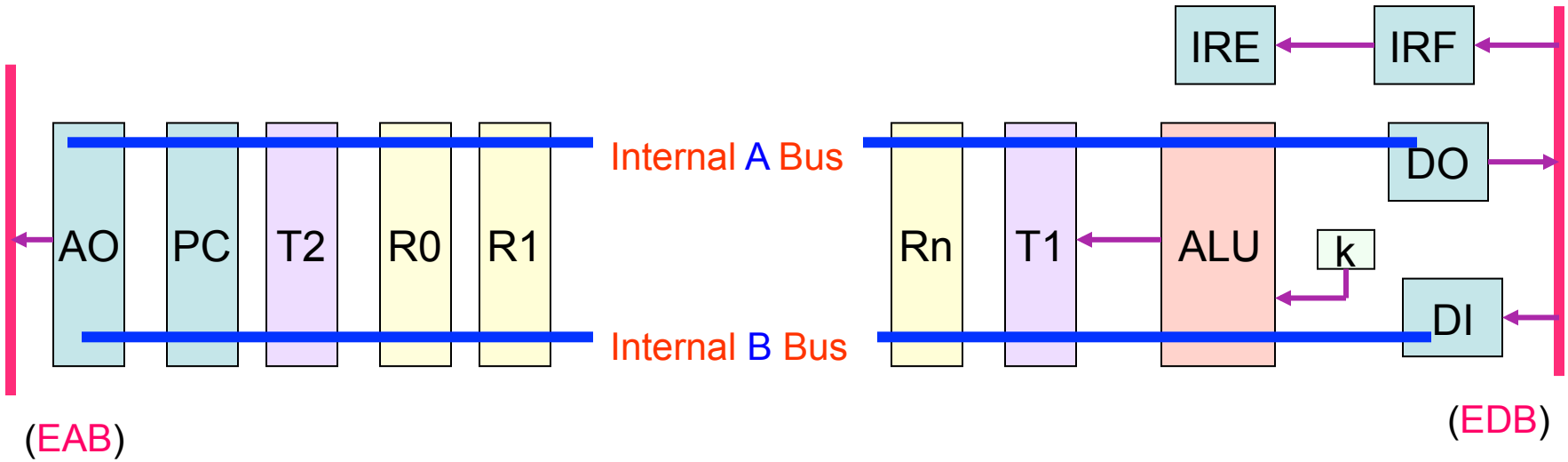
OP	TY	NA

MIN Control Word

Control Fields

AO	PC	T2	Regs	T1	ALU	K	DI	DO	IRE	IRF	….

MIN Execution Unit



Control Word Decoder

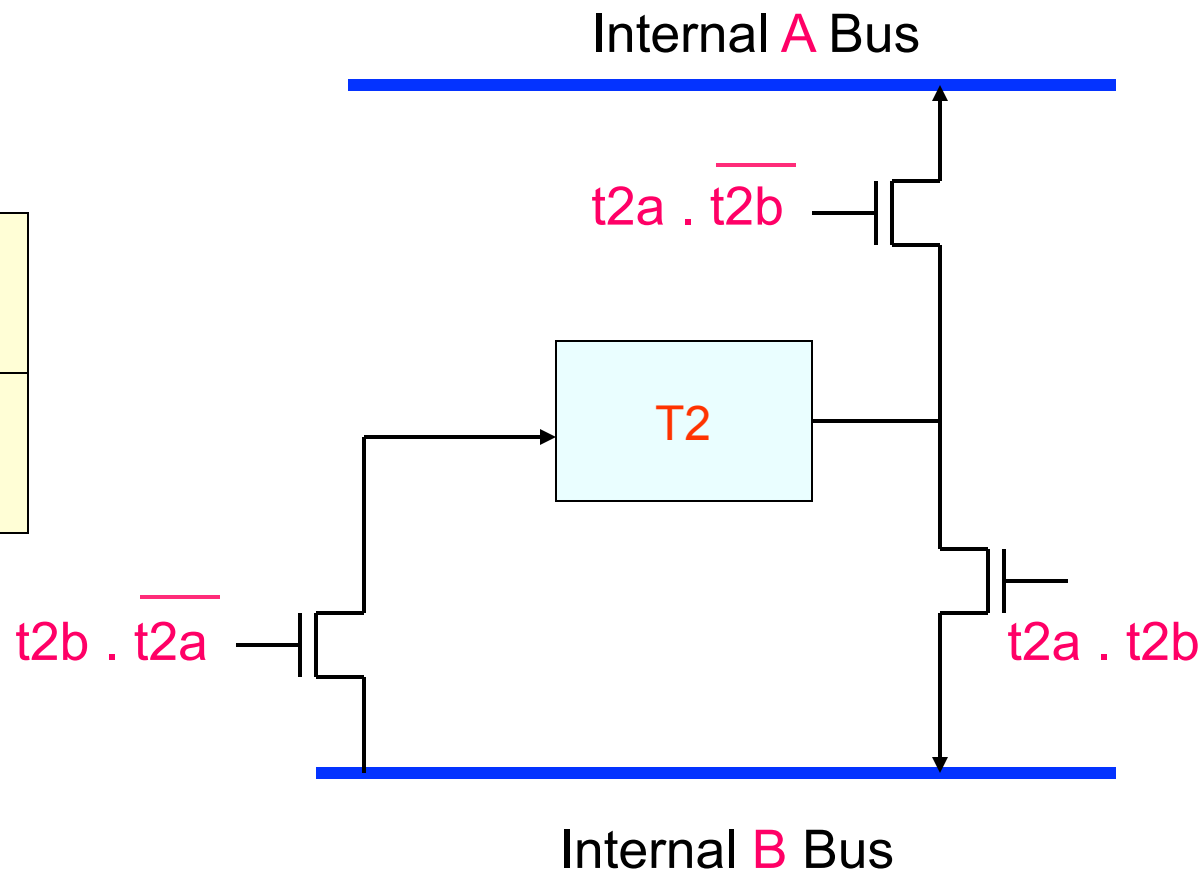
- How many bits each control needs?
- Procedure
  1. List uses of the macro
  2. Allocate bits
  3. Use a Karnaugh map to assign bit patterns
- Collect all the occurrences (PC, T2, RX …)
- Assign no. of bits to control fields

Control Word Decoder

- T2 Control
  - T2 occurrences
    - t2 → a
    - t2 → b
    - a → t2 (only one occurrence – abdm4)
    - b → t2
    - None
  - Assign two bits (4 occurrences)

T2 Control

t2b \ t2a	0	1
0	none	t2 → a
1	b → t2	t2 → b

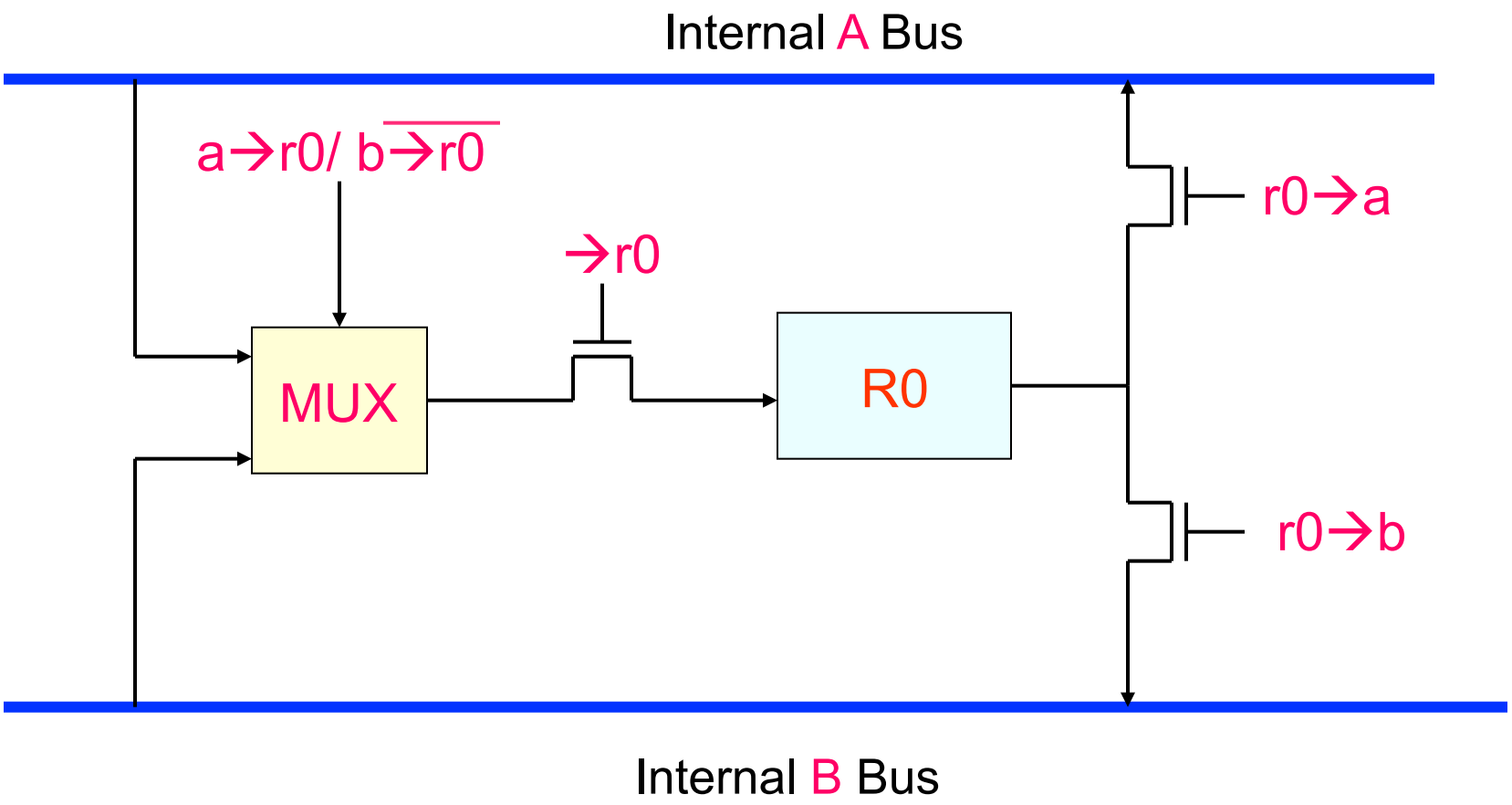


Control Word Decoder

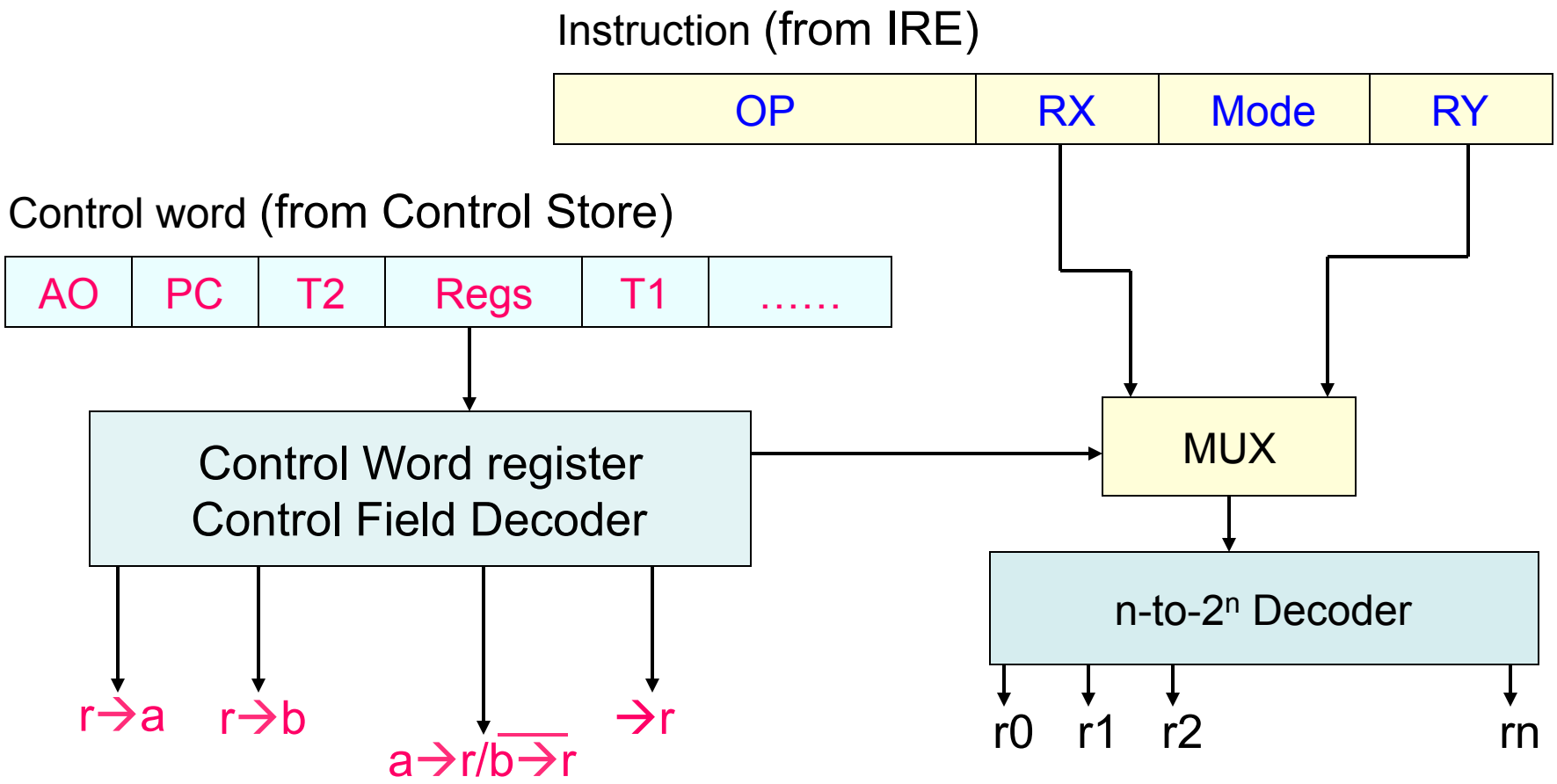
- Register control
  - RX and RY occurrences

ry → a	rx → a; ry → b
b→ rx	b→ ry
ry → b; b → rx	b → rx; a → ry
rx → a	rx → a; b → ry
rx → b; b → ry	none

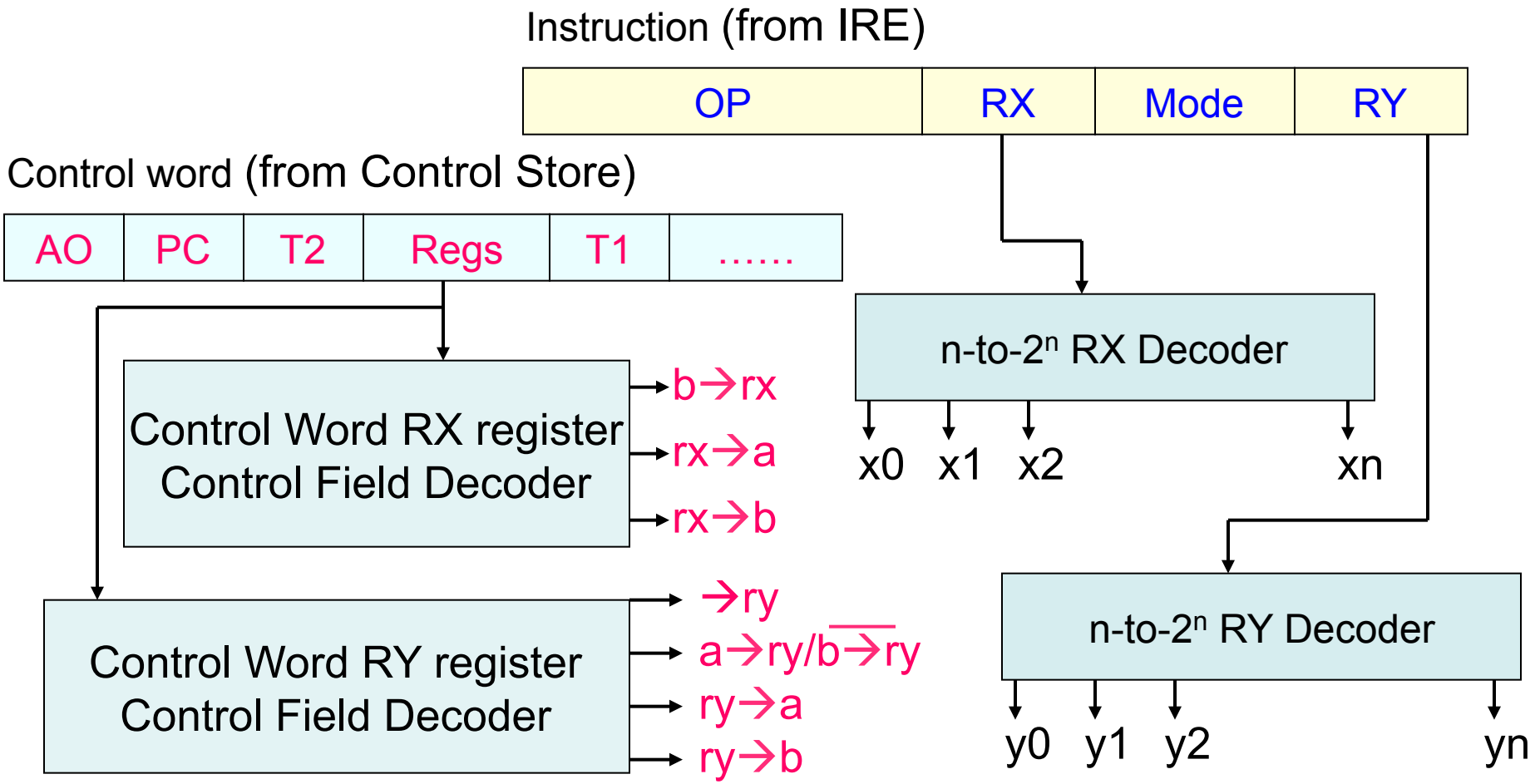
Register Control



Register Control



Register Control



Control Word Decoder

- Register control

→ r0 = (b → rx).x0 + (→ ry). y0	load r0
a→ r0 = (b → rx).x0 + ((a → ry)/(b → ry)'). y0	load from A
r0→ a = (rx → a).x0 + (ry → a). y0	to A
r0→ b = (rx → b).x0 + (ry → b). y0	to B

Control Word Decoder

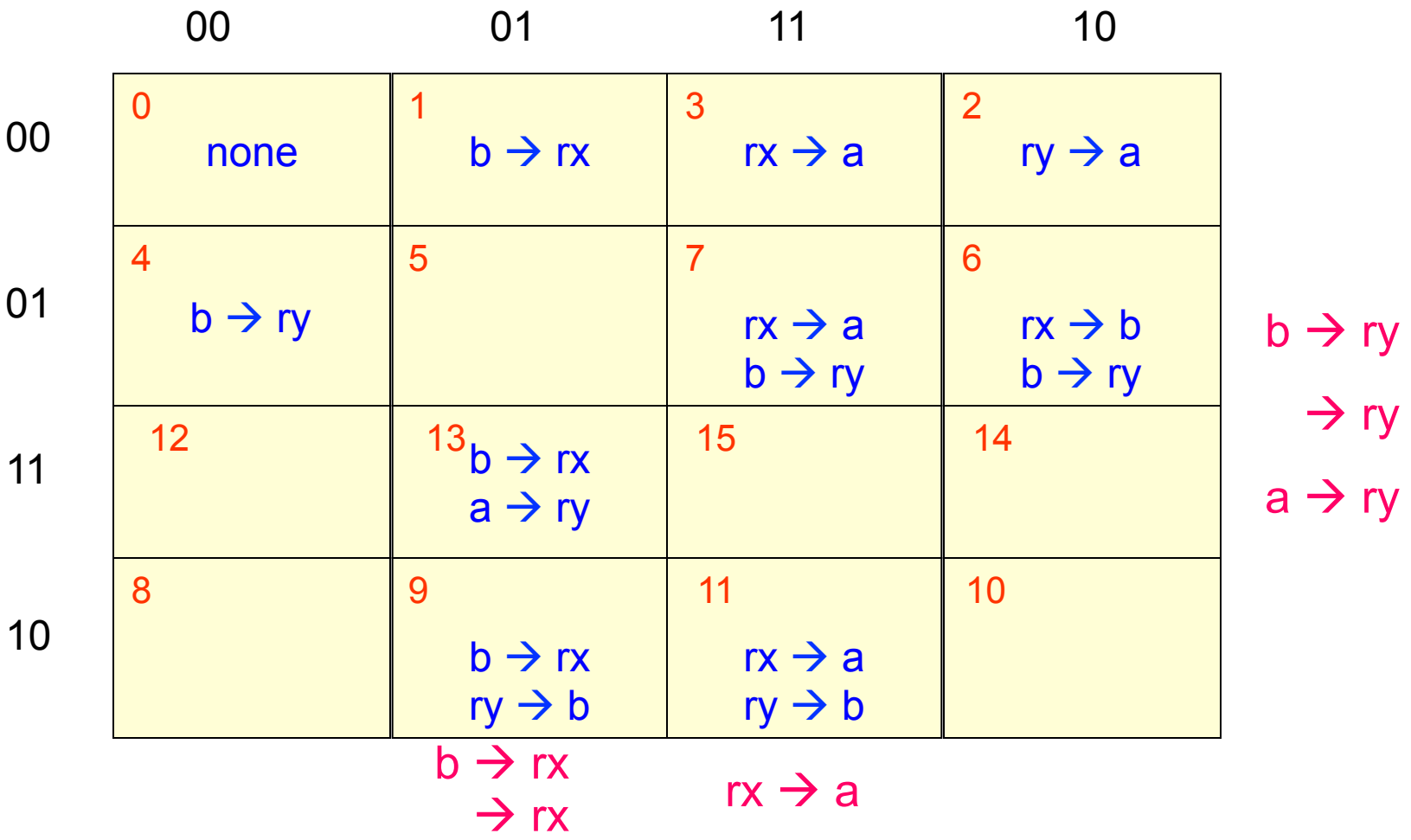
- Control Word States
  - ry → a
  - b → rx
  - ry → b; b → rx
  - Rx → a
  - Rx → b; b → ry
  - rx → a; ry → b
  - b → ry
  - b → rx; a → ry
  - rx → a; b → ry
  - none
- Control Lines
  - ry → a
  - b → rx; → rx
  - ry → b; → rx; b → rx
  - rx → a
  - rx → b; → ry; b → ry
  - rx → a; ry → b
  - b → ry; → ry
  - b → rx; → ry; a → ry; → ry
  - rx → a; → ry; b → ry
  - none

Control Word Decoder

- Control Lines

rx → a
ry → a
rx → b
ry → b
→ rx
→ ry
a → ry
b → rx
b → ry

Register Control



Control Word Decoder

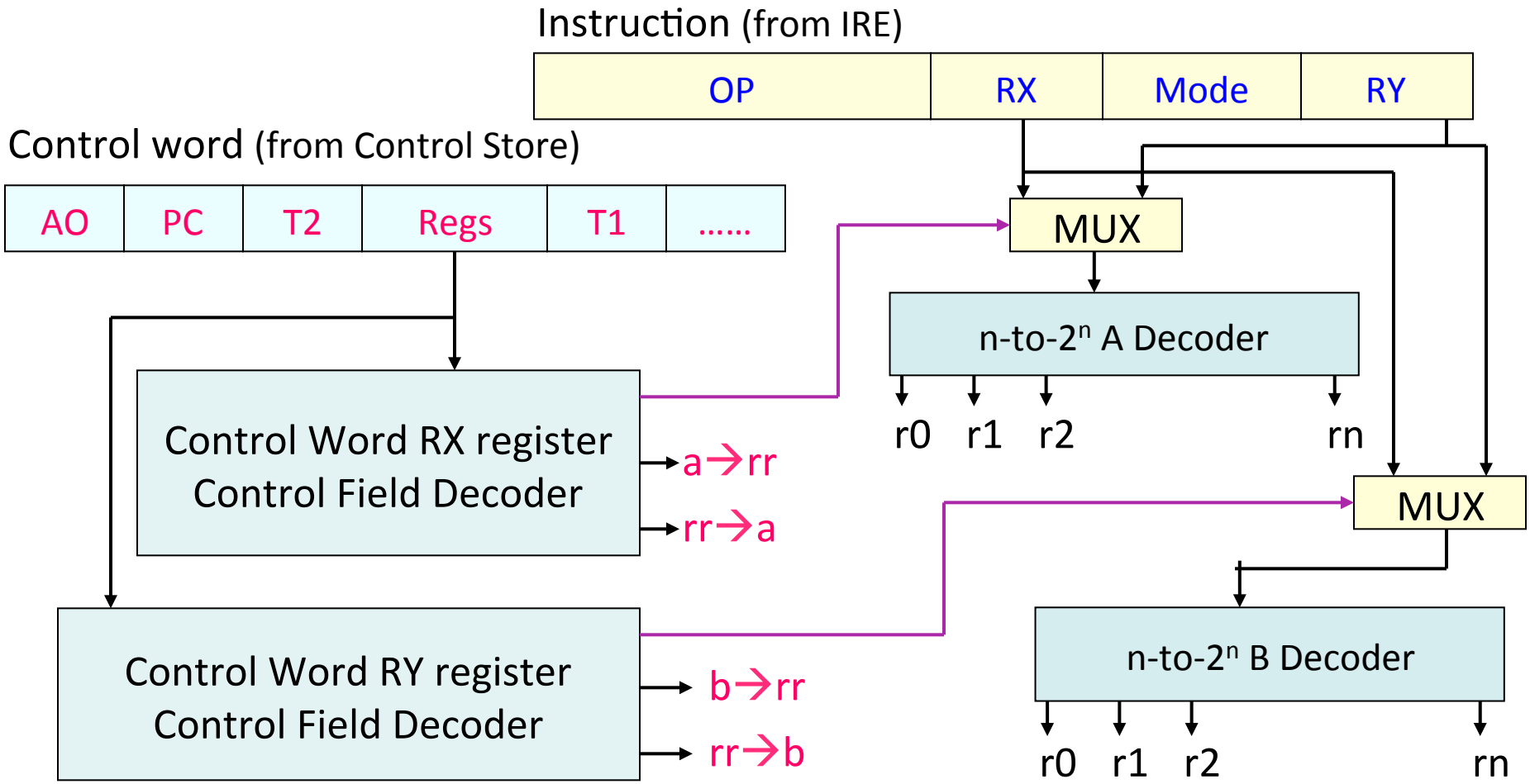
Control Word States	Control Bit Assignment
None	0000
b → rx	0001
ry → a	0010
rx → a	0011
b → ry	0100
rx → b; b → ry	0110
rx → a; b → ry	0111
ry → b; b → rx	1001
rx → a; ry → b	1011
b → rx; a → ry	1101

Control Word Decoder

- Control Lines
- Decoder Patterns

Control Lines	Decoder Patterns
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

Register Control



ALU Control

Control Word state

a→ alu; +1 → alu; add-n; alu→ t1

a→ alu; b → alu; add-n; alu→ t1

a→ alu; 0 → alu; add-s; alu→ t1

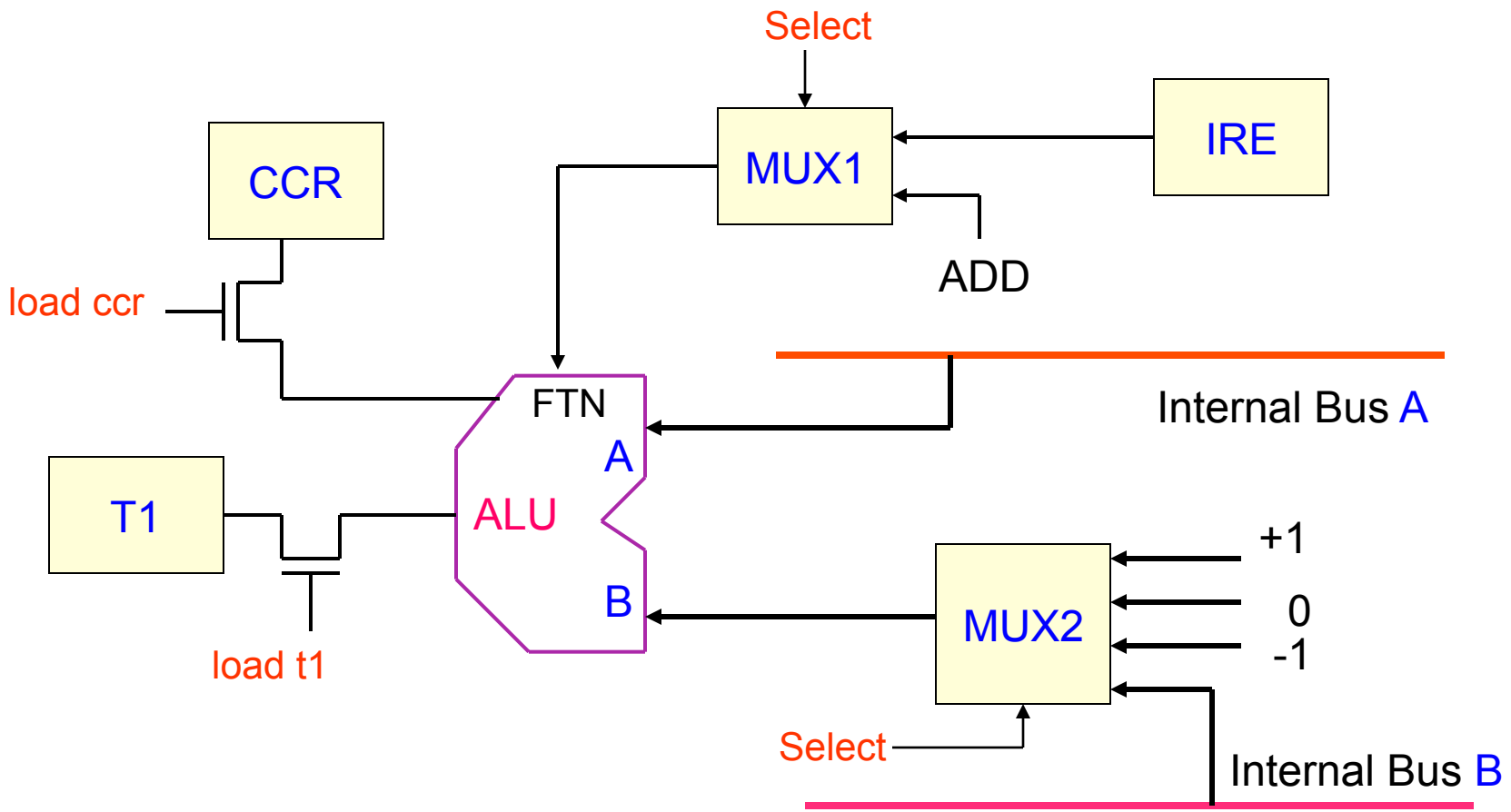
a→ alu; b → alu; op-s; alu→ t1

a→ alu; -1→ alu; add-n; alu→ t1

Control Lines

- load t1
- load ccr
- add/op select
- alu-b input select

ALU Control (1/4)



ALU Control (2/4)

1 \ 0	00	01	11	10
0	0 none	1	3	2 b→alu add-n
1	4 0 → alu add-s	5 +1 → alu add-n	7 -1 → alu add-n	6 b → alu op-s

Karnaugh map for ALU control assignment

ALU Control (3/4)

Control Word state

a→ alu; +1 → alu; add-n; alu → t1

a→ alu; b → alu; add-n; alu→ t1

a→ alu; 0 → alu; add-s; alu→ t1

a→ alu; b → alu; op-s; alu→ t1

a→ alu; -1→ alu; add-n; alu→ t1

none

Control Field bit assignment

- 101
- 010
- 100
- 110
- 111
- 000

ALU Control (3/4)

Control Lines

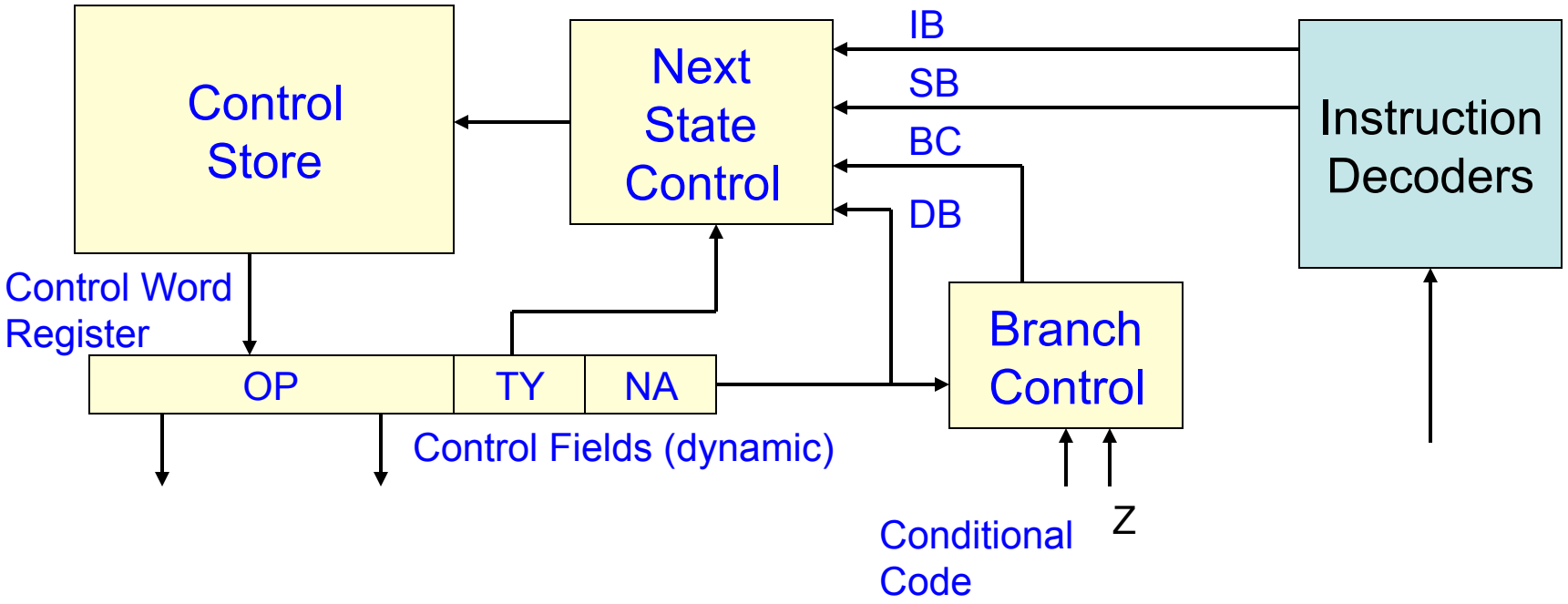
- load t1
- load ccr
- add/op select
- alu-b input select

Decoder Patterns

- xxx | 000 (all except 000)
- 1x0
- 110
- x10

	00	01	11	10
0	0 none	1	3	2 b→alu add-n
1	4 0 → alu add-s	5 +1 → alu add-n	7 -1 → alu add-n	6 b → alu op-s

Next State Logic



Thank You