CISC Design

Hardware Flowchart

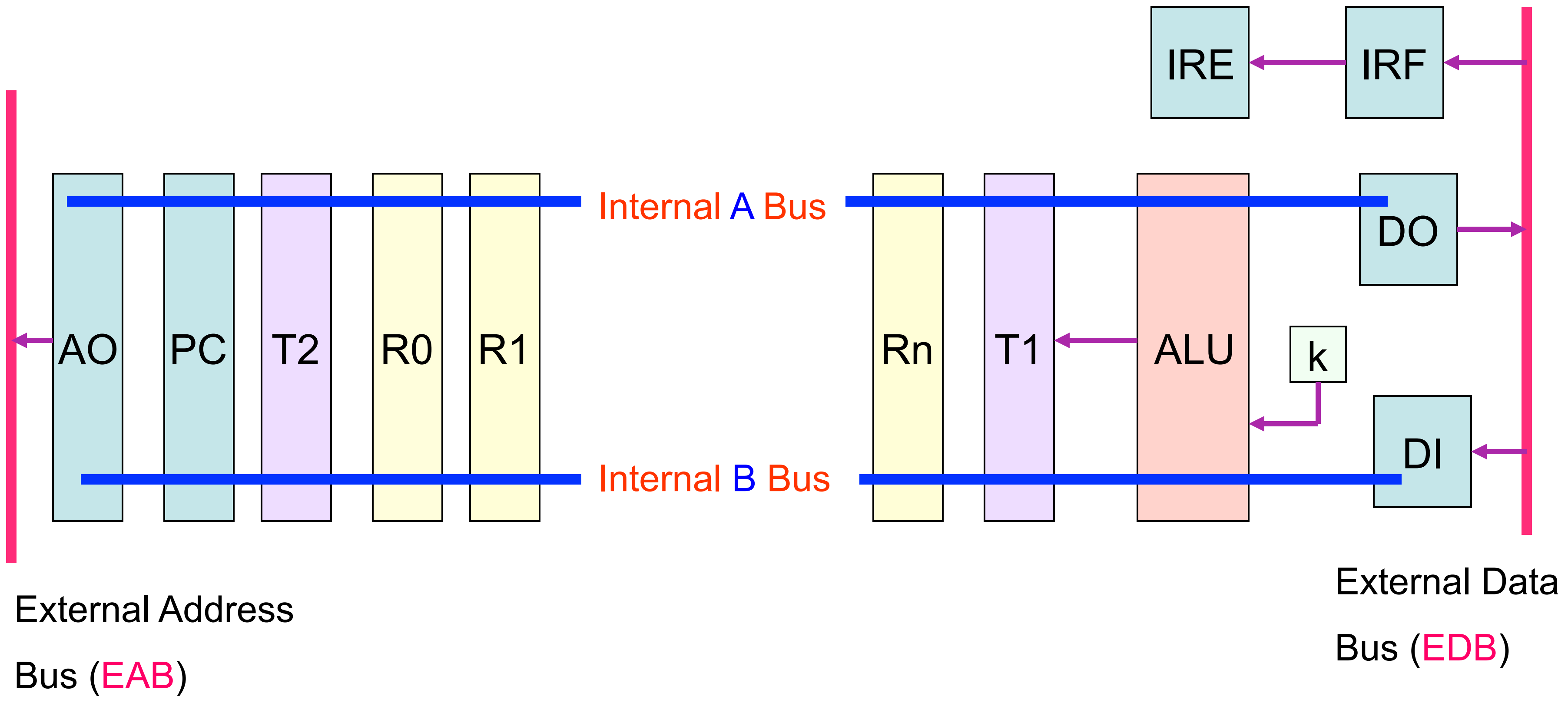
Virendra Singh
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-309: Microprocessors

Lecture 25 (22 Sep 2015)

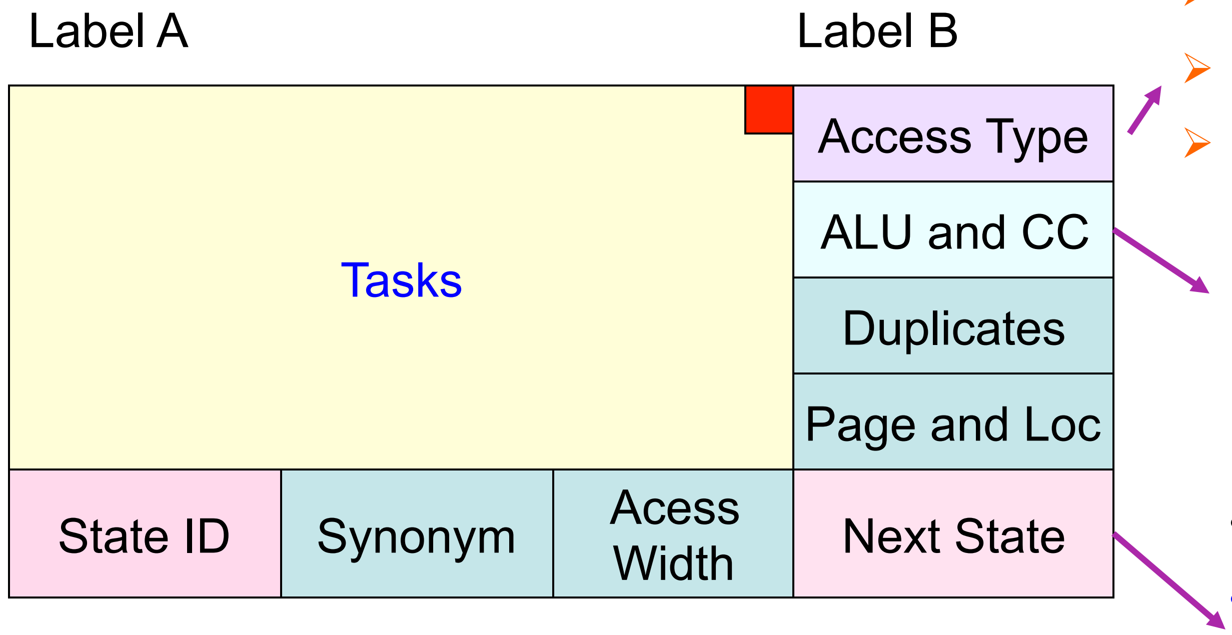
CADSL

MIN Datapath



Level 2 Flowcharts

Format for Level 2 flowchart state



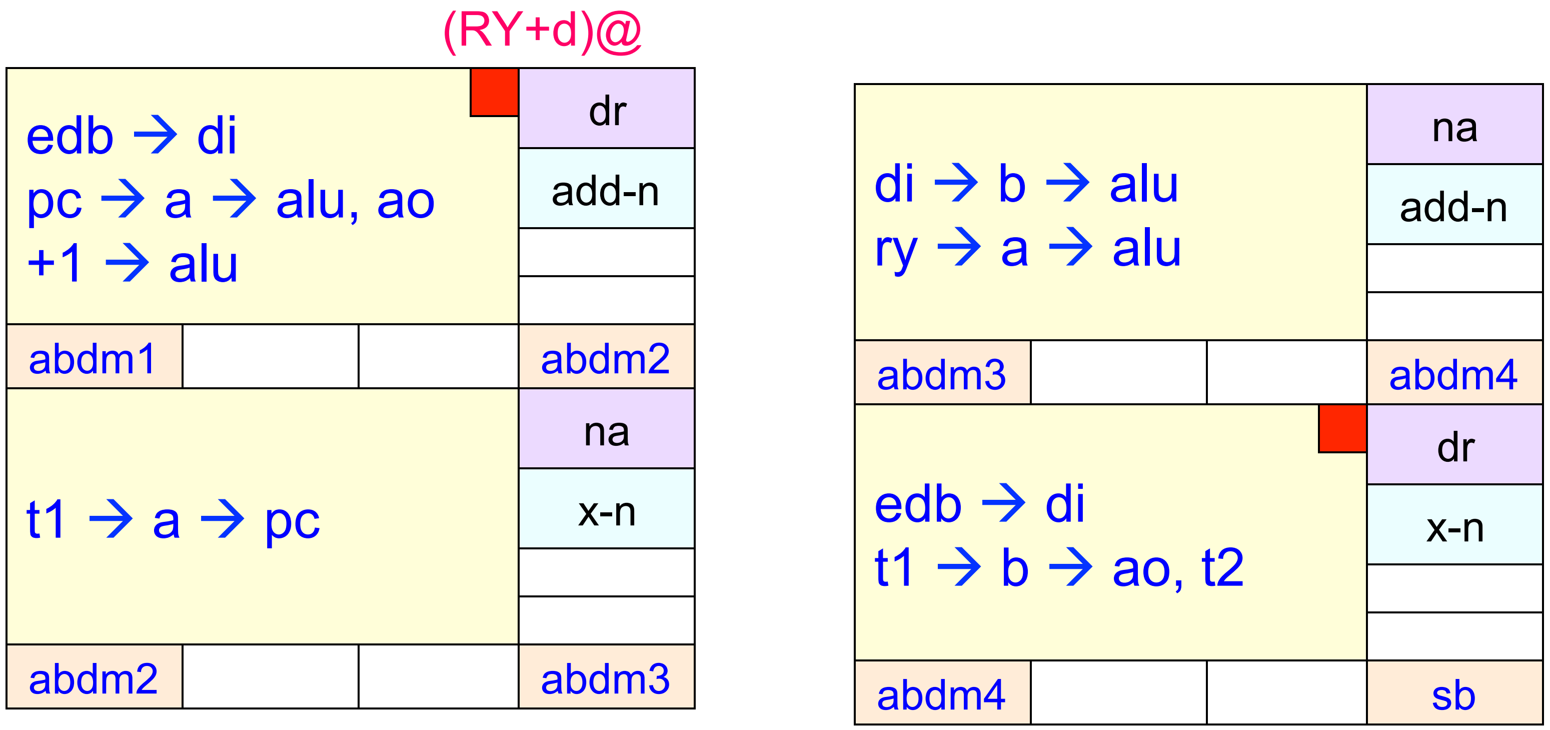
- DR Data Read
- DW – Data Write
- IR - Instruction Read
- NA – No access

- S – Set
- N – Not set
- X – Don't care

- BC – Branch conditionally
- IB – Instruction branch
- SB – Sequence branch
- StateID – Direct branch

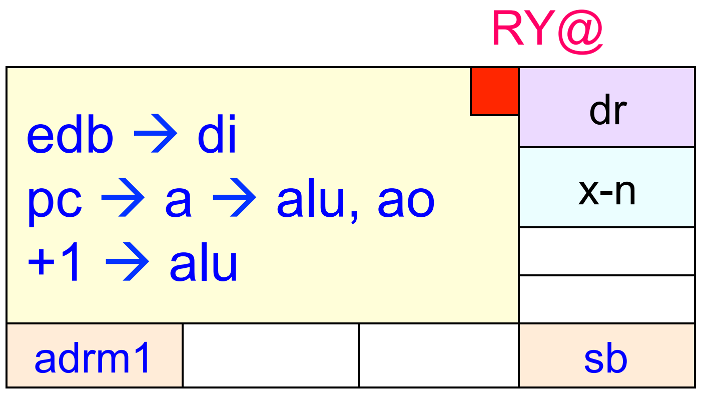
Level 2 Flowchart: Address Mode Sequences

Base Plus Displacement



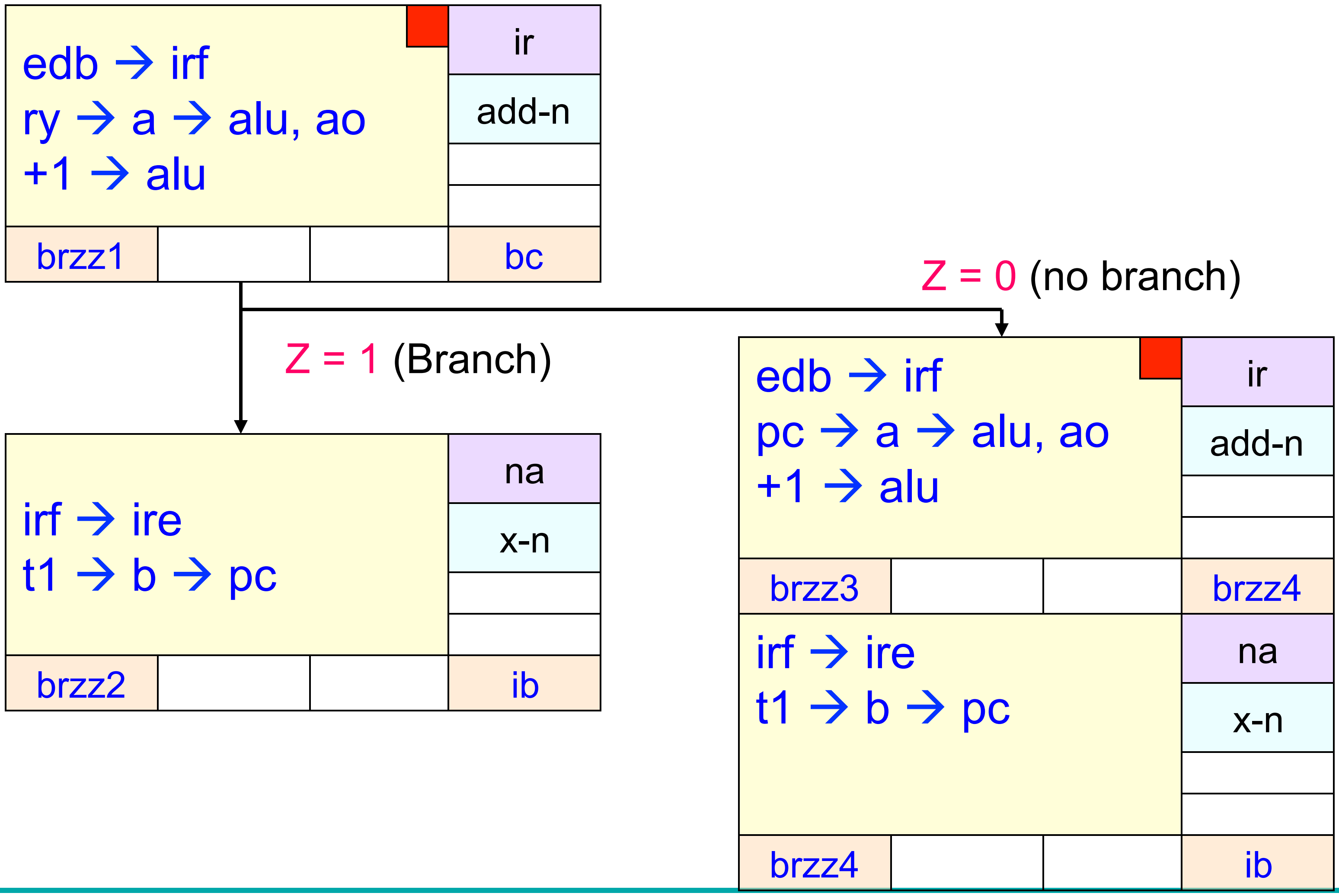
Level 2 Flowchart: Address Mode Sequences

Register Indirect



Level 2 FC: Address Mode Sequences

Branch Instruction



Level 2 Flowchart: Execution Sequences

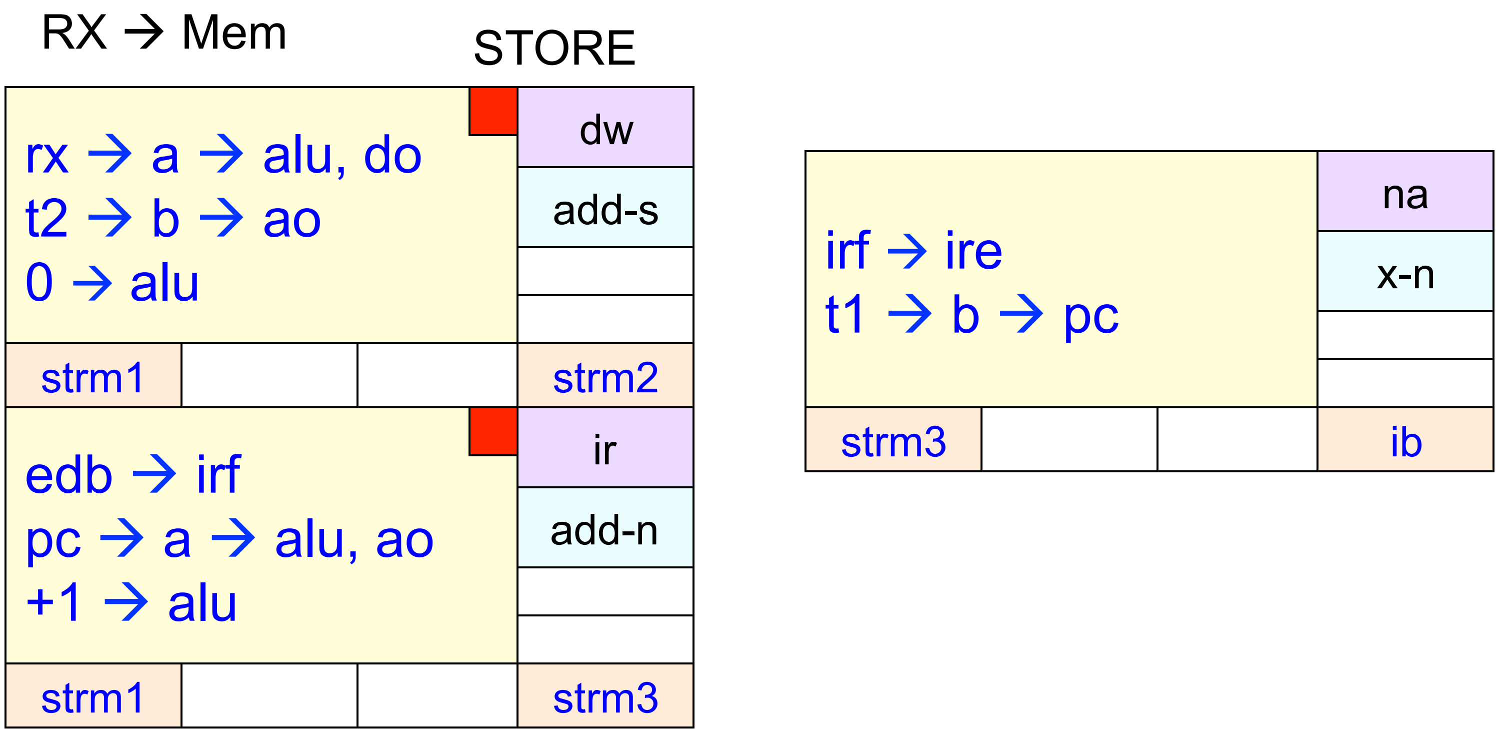
Execution sequences with memory operand reference

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

Level 2 Flowchart: Execution Sequences

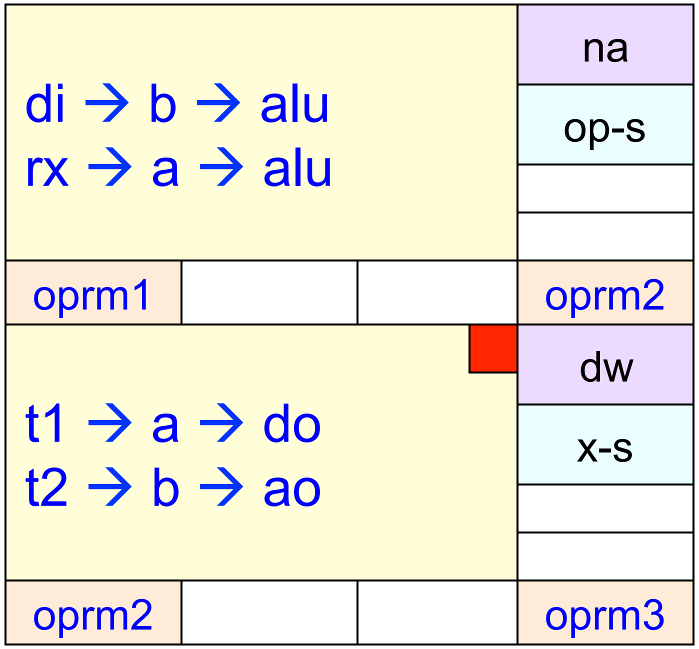
Execution sequences with memory operand reference

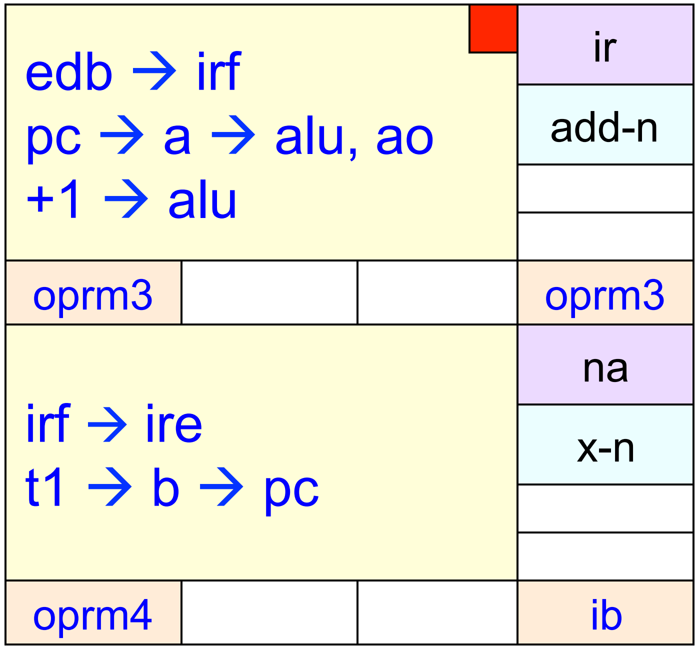


Level 2 Flowchart: Execution Sequences

Execution sequences with memory operand reference

RX OP Mem → Mem   ADD, AND, SUB





Level 2 Flowchart: Execution Sequences

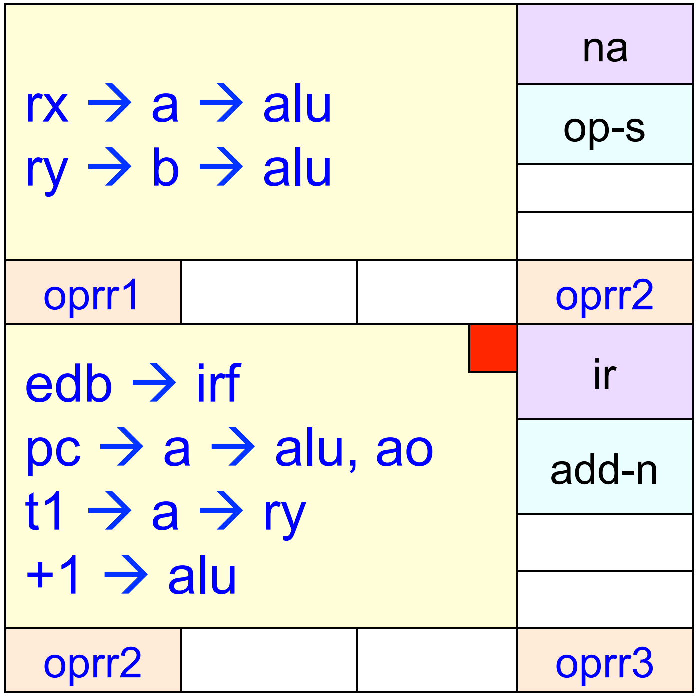
Execution sequences with memory operand reference

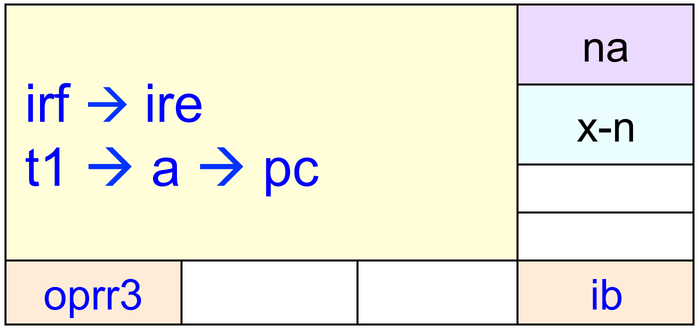
Mem → ALU			TEST
di → b → t2 edb → irf pc → a → alu, ao +1 → alu		🟥	ir add-s
test1			test2
irf → ire t1 → b → pc t2 → a → alu 0 → alu			na add-s
test2			ib

Level 2 Flowchart: Execution Sequences

Execution sequences for Register-to-Register and special instructions

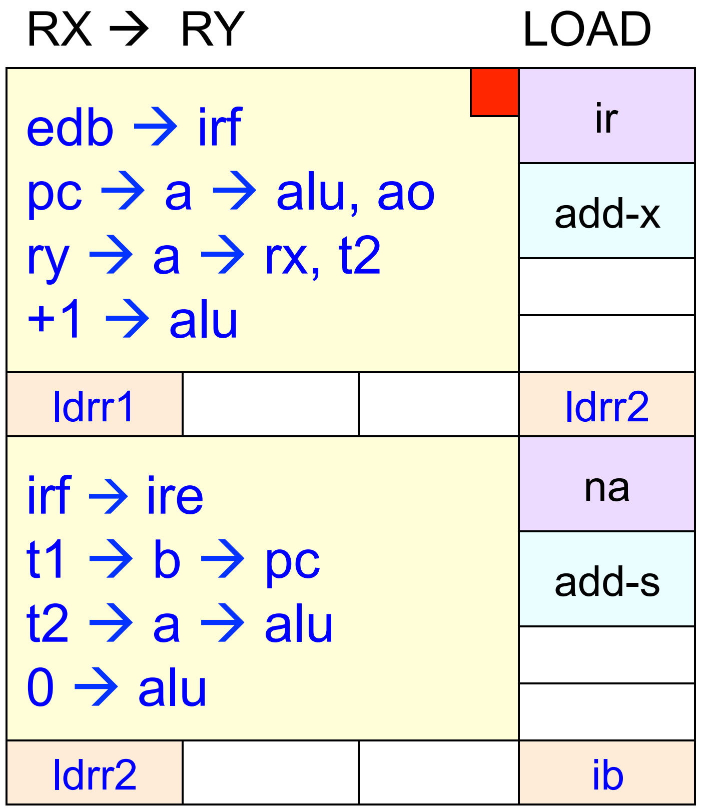
RX OP RY → RY   ADD, SUB, AND

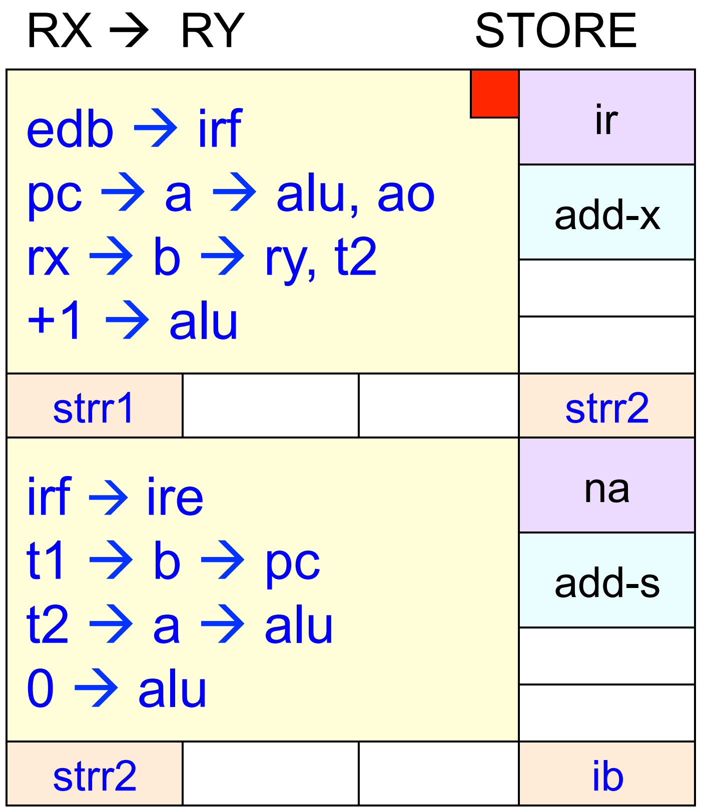




Level 2 Flowchart: Execution Sequences

Execution sequences for Register-to-Register and special instructions





Level 2 Flowchart: Execution Sequences

Execution sequences for Register-to-Register and special instructions

RY@ → RX
RY+1 → RY

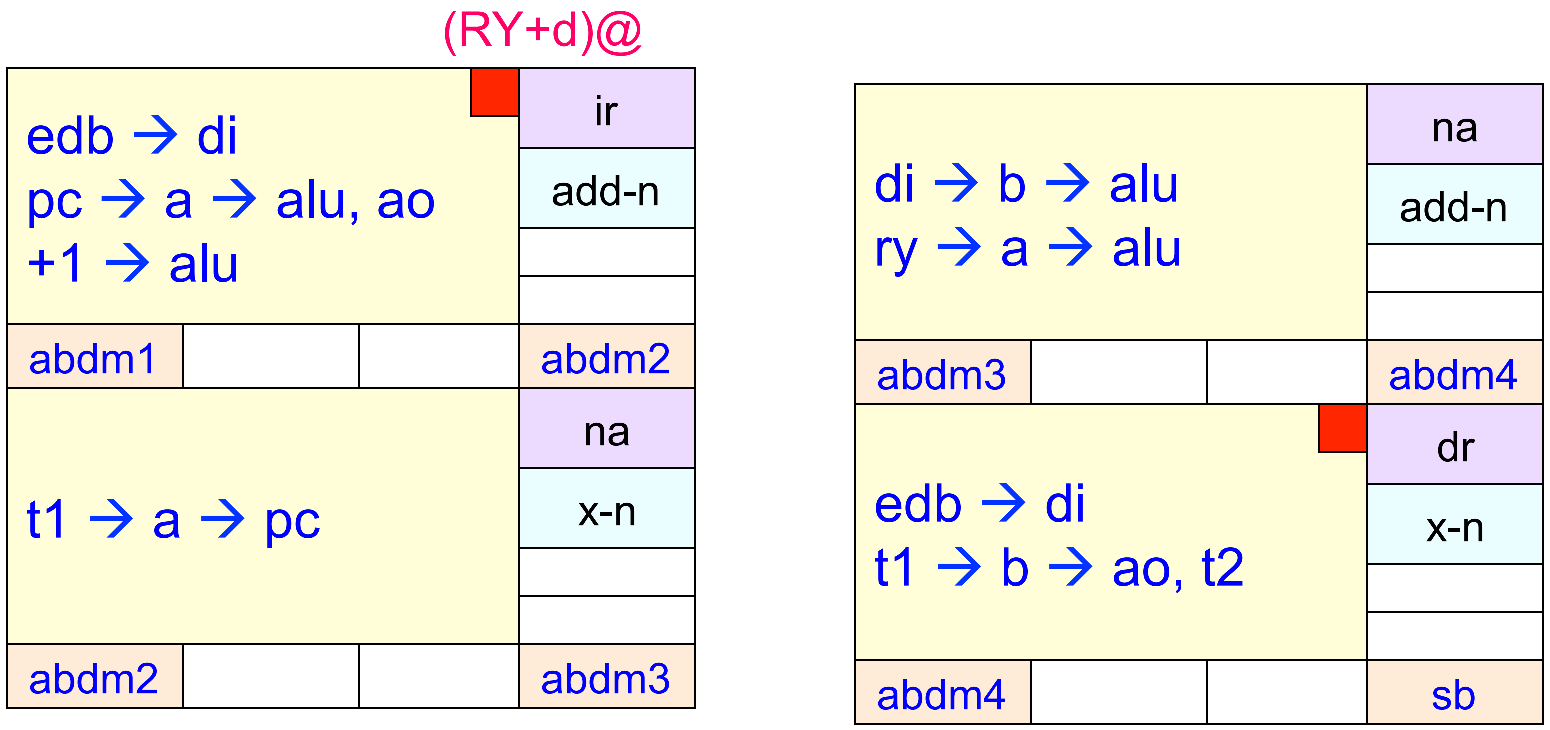
POP

edb → di ry → a → alu, ao +1 → alu			dr add-n
popr1			popr2
di → b → rx t1 → a → ry			na x-n
popr2			popr3

edb → irf pc → a → alu, ao +1 → alu			ir add-n
popr3			popr4
irf → ire t1 → a → pc			na x-n
popr4			ib

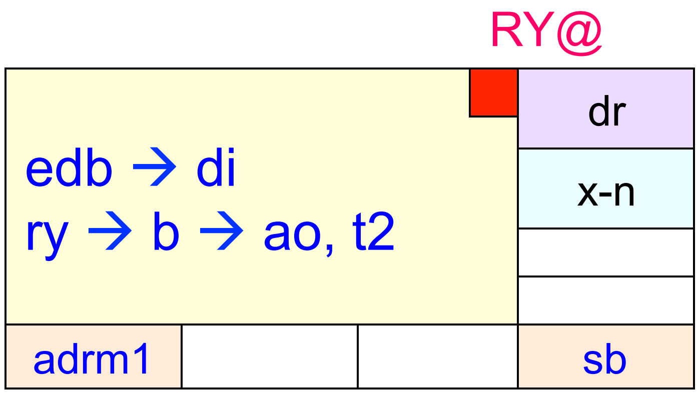
Merged Level 2 Flowchart: Address Mode Sequences

Base Plus Displacement



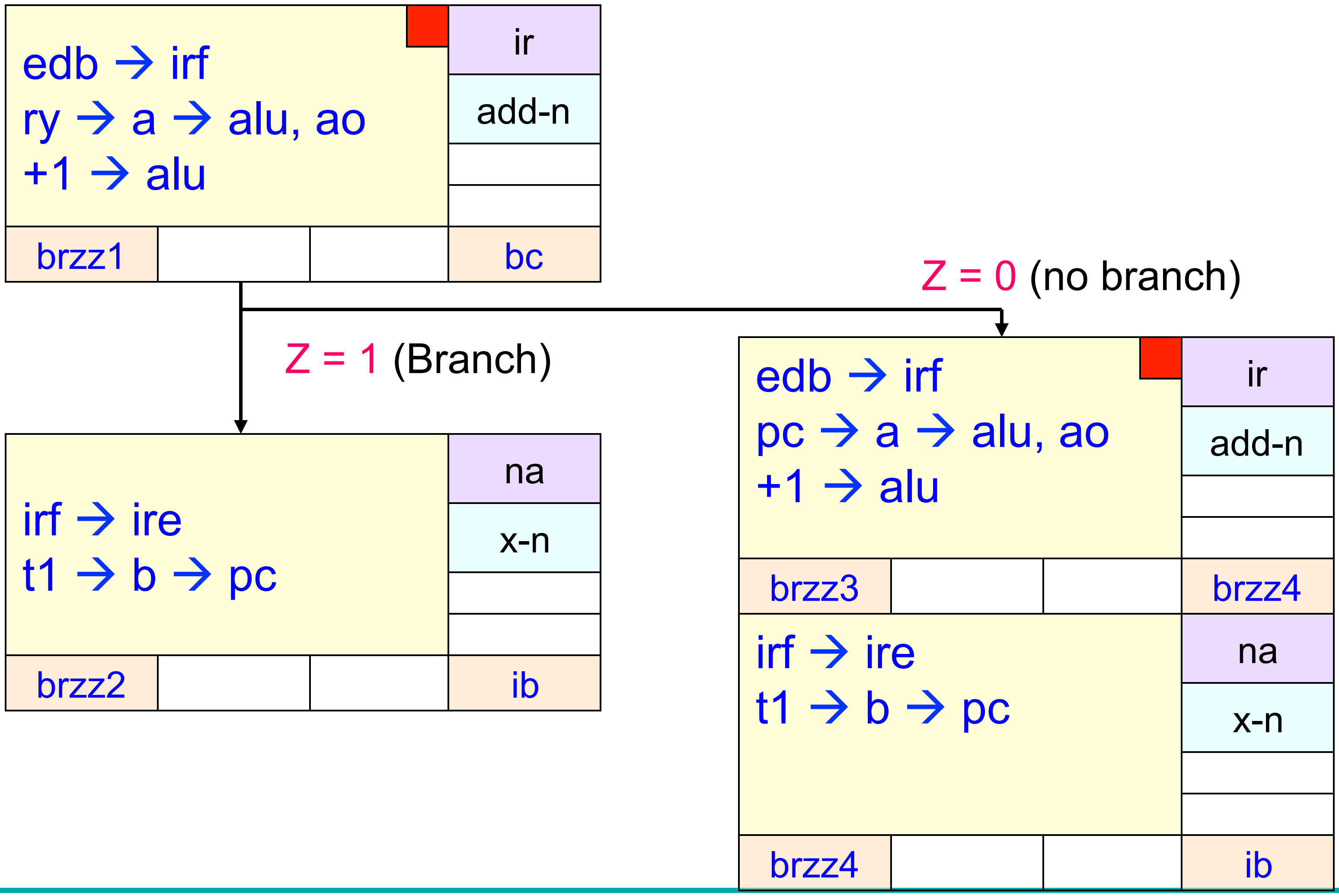
Merged Level 2 Flowchart: Address Mode Sequences

Register Indirect



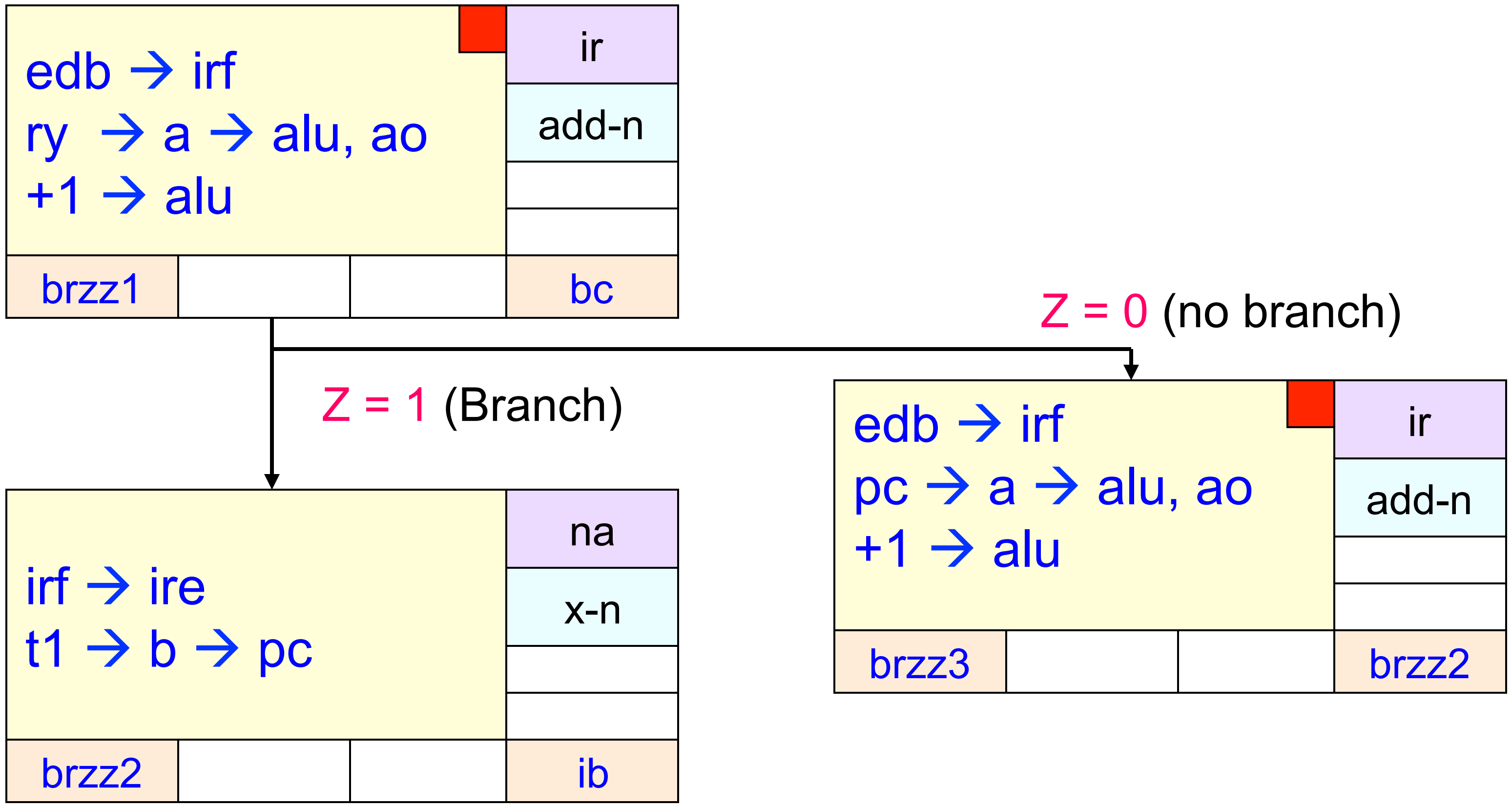
How to Merge Level 2 FC?

Branch Instruction



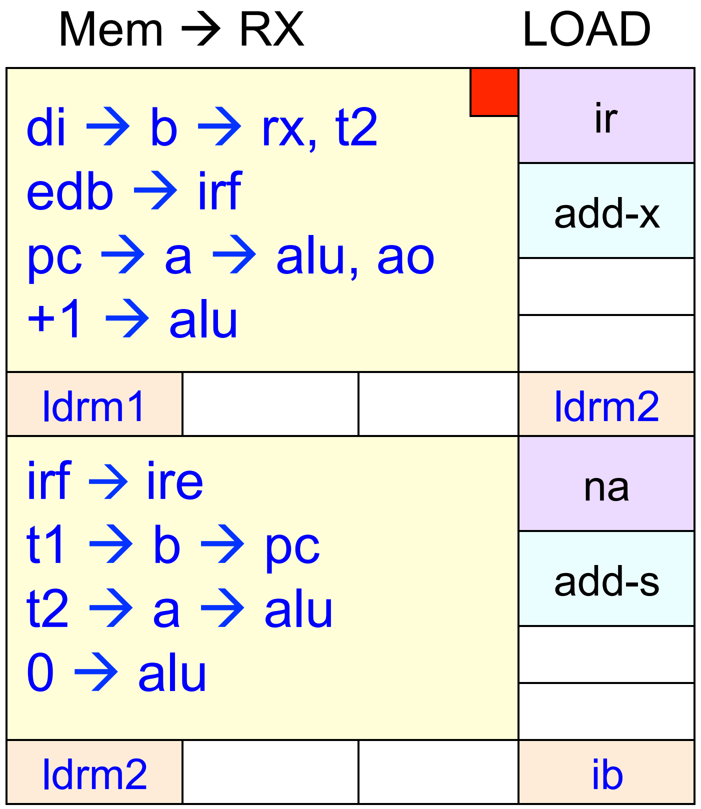
Merged Level 2 Flowchart: Address Mode Sequences

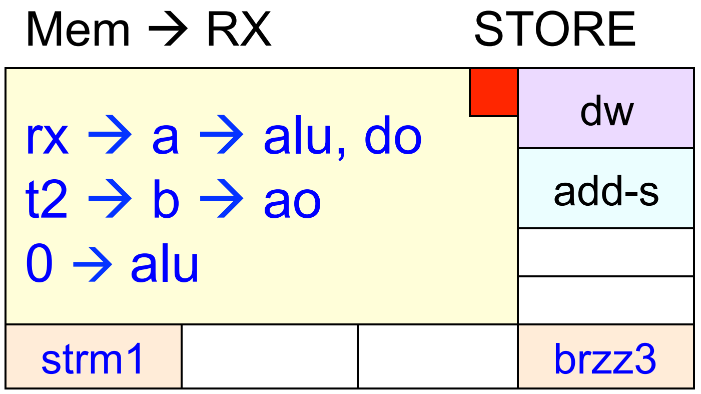
Branch Instruction



Merged Level 2 Flowchart: Execution Sequences

Execution sequences with memory operand reference

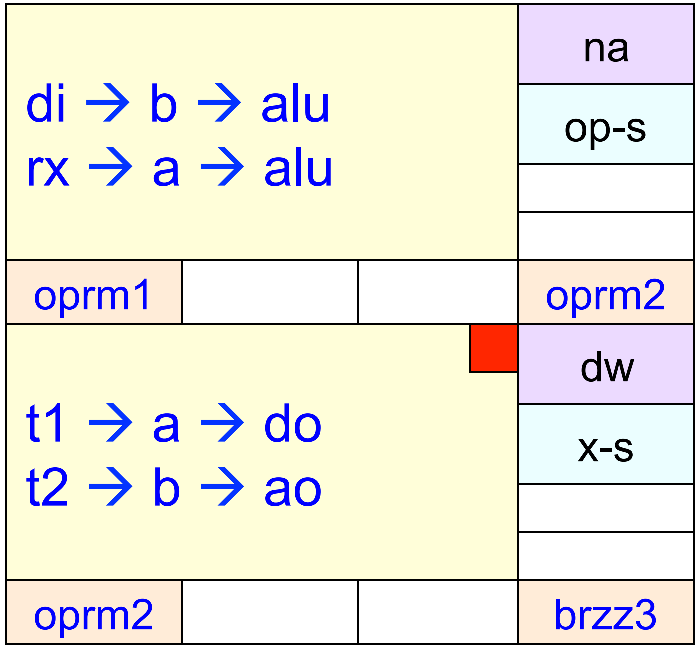




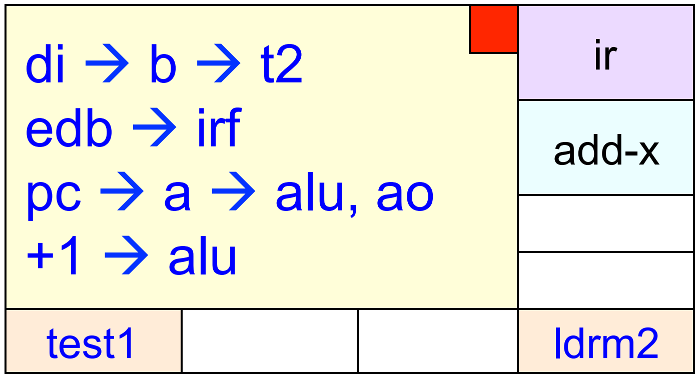
Merged Level 2 Flowchart: Execution Sequences

Execution sequences with memory operand reference

RX OP Mem → Mem   ADD, AND, SUB



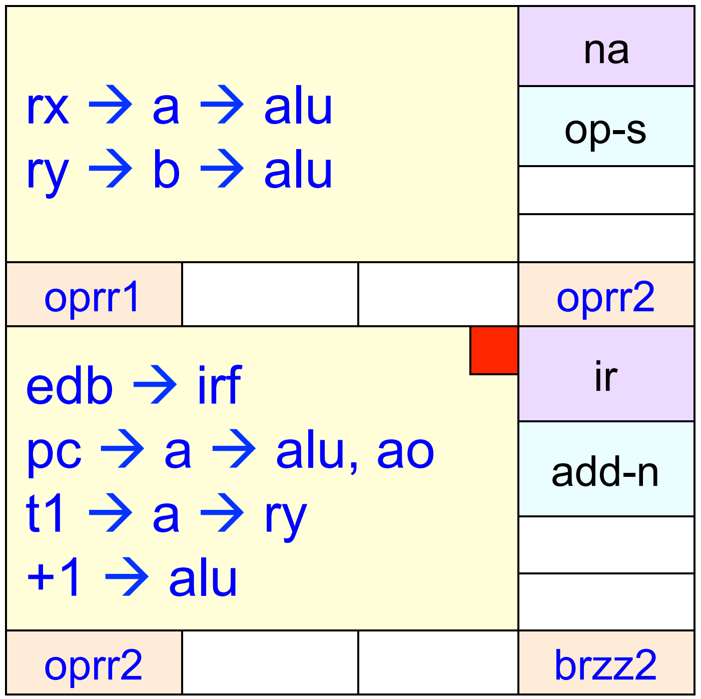
Mem → ALU   TEST



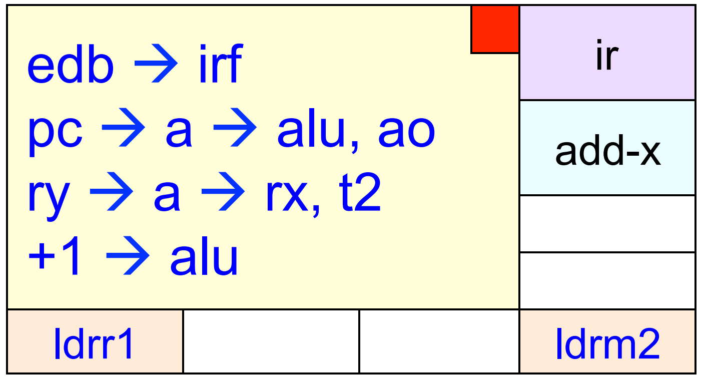
Merged Level 2 Flowchart: Execution Sequences

Execution sequences for Register-to-Register and special instructions

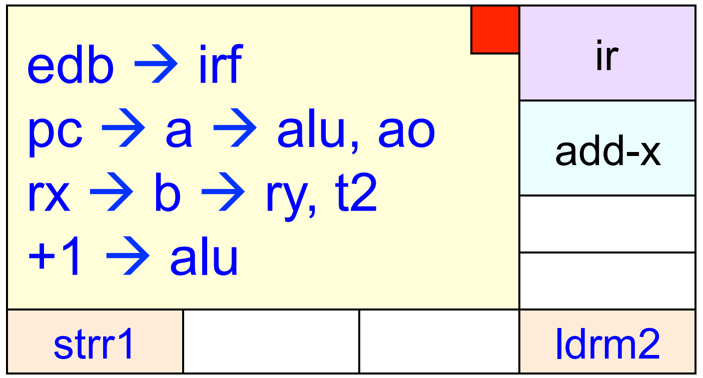
RX OP RY → RY ADD, SUB, AND



RX → RY LOAD



RX → RY STORE



Merged Level 2 Flowchart: Execution Sequences

Execution sequences for Register-to-Register and special instructions

RY@ → RX
RY+1 → RY

POP

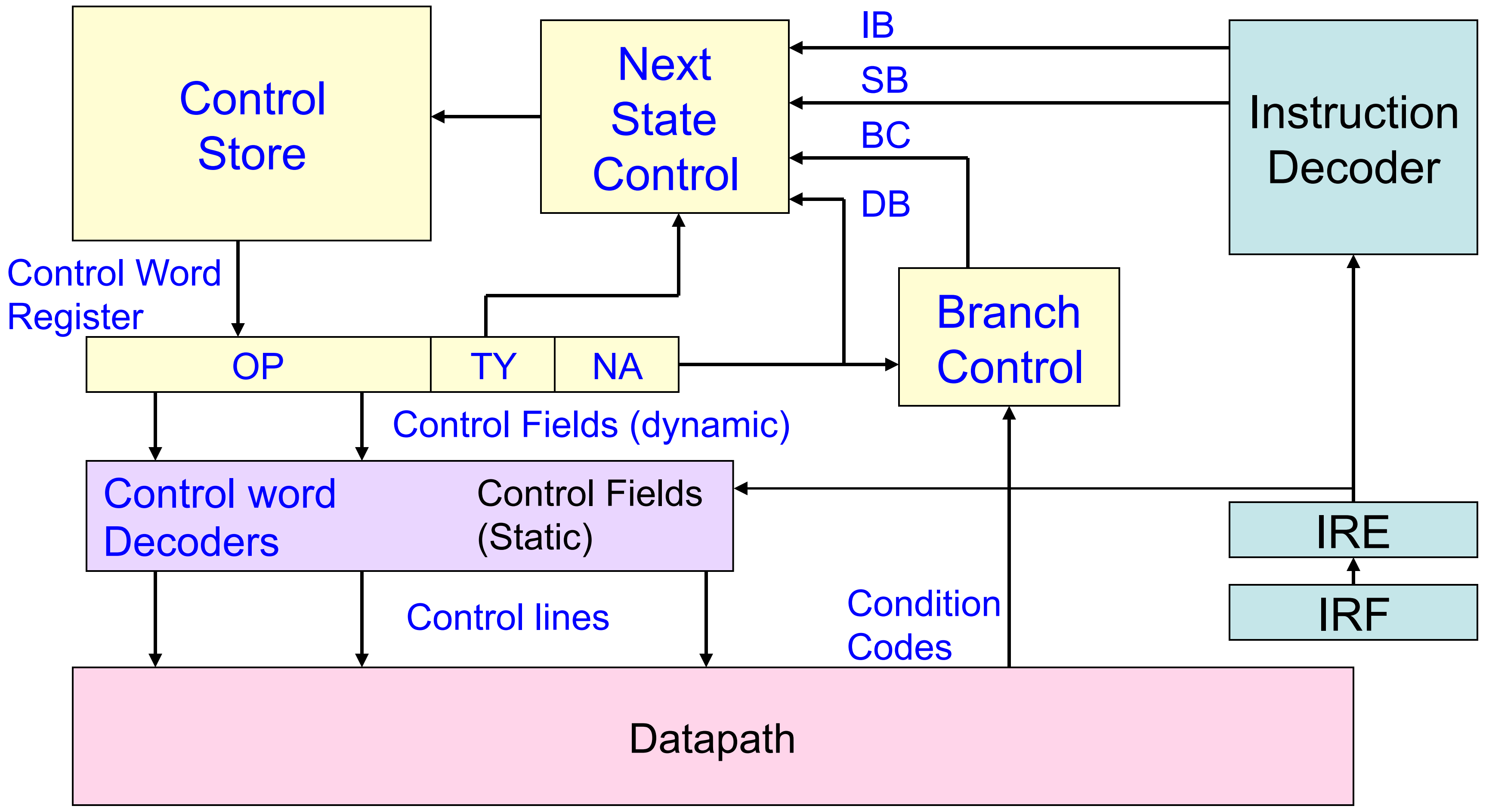
edb → di ry → a → alu, ao +1 → alu		🟥	dr
			add-n
popr1			popr2
di → b → rx t1 → a → ry			na
			x-n
popr2			brzz3

RY-1 → RY
RY@ → RX

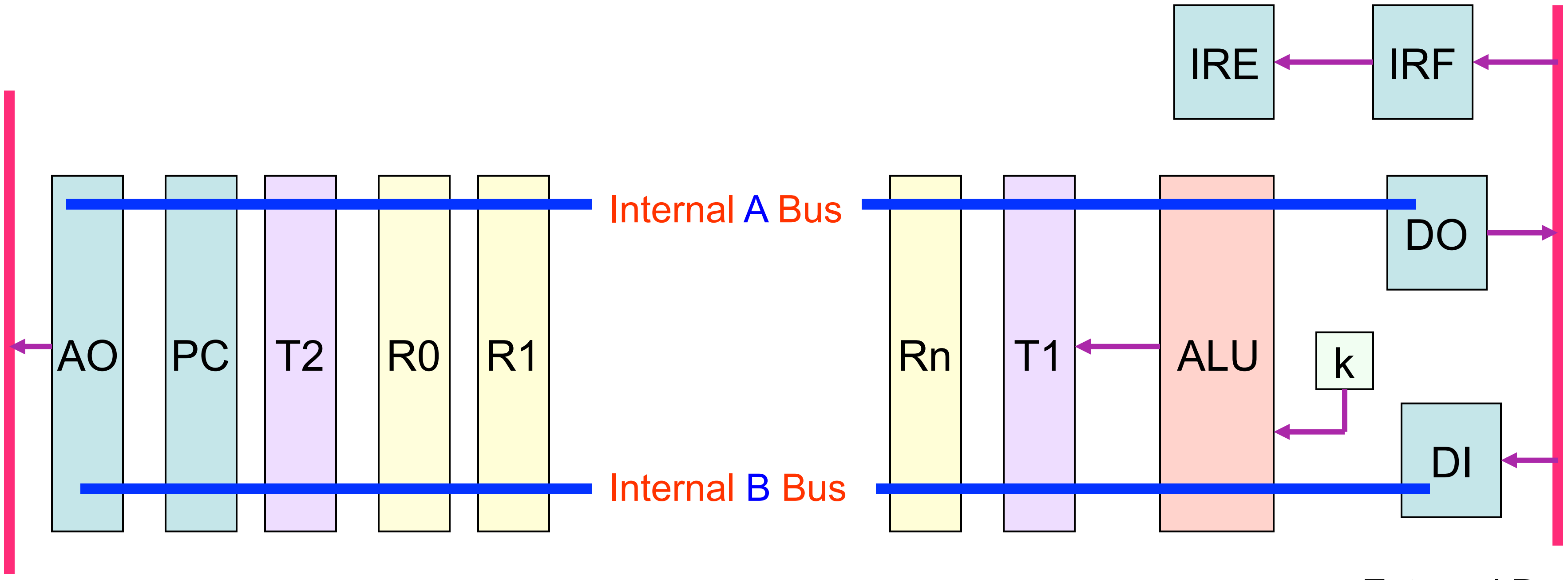
PUSH

ry → a → alu -1 → alu			na
			add-n
push1			push2
rx → a → do t1 → b → ao, ry		🟥	dw
			x-n
push2			brzz3

Processor Block Diagram



MIN Datapath



External Address

Bus (EAB)

External Data

Bus (EDB)

Implementation

- Each state in Level 2 flowchart corresponds to one control word
- Transformation of flowcharts into control store bit patterns
  - The task become bits in the control fields (OP)
  - The next state becomes in the control store address select (TY) and next address (NA)
  - The state ID becomes the location of the control word in the control store

Implementation

Relationship between Flowcharts and Hardware

- Flowchart – compact and precise description of hardware requirements
- Steps for implementing microcoded controller

1. Execution Unit
   - Develop concurrently
   - Add things as and when needed

Implementation

2. Instruction Decoders

- Translate an instruction bit pattern to the control store address for the execution sequence
- Two decoders are needed (for MIN)
- First, translates the instruction bit pattern into the control store address for the appropriate address mode sequence (provide IB)
- Second, translates the instruction bit pattern into control store address for the execution sequence (provide SB)

Implementation

3. Control word format
    - Derived from flowcharts
    - HFC can tell required capability of control word precisely

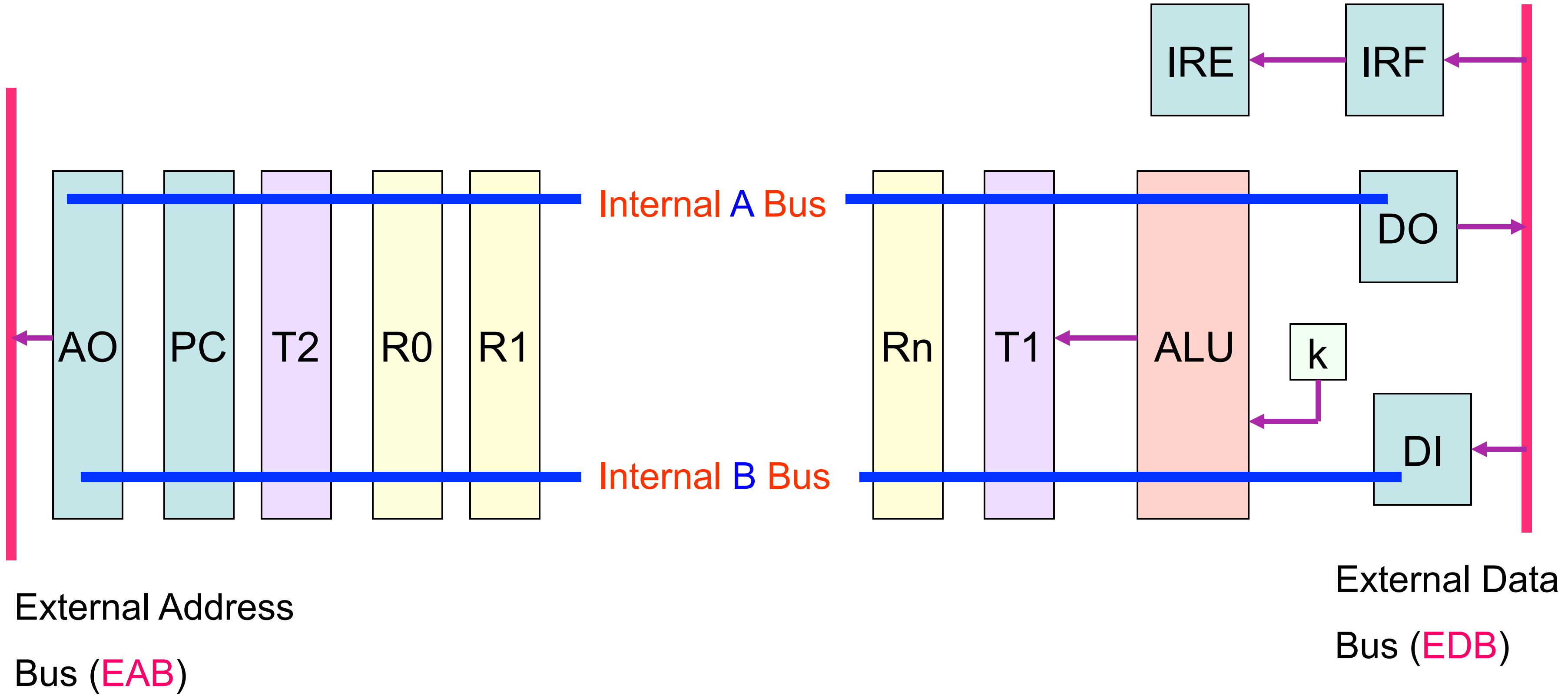
5. Control word decoders
    - Combine control word (dynamic) control fields, the IRE (static) control fields, and timing signals, to provide the gate control signals for all transfers in the Datapath and the Controller

6. Controller block diagram

Implementation

- Design of flowchart
  - Made some assumptions (buses, registers..)
  - Collect the assumptions and implement

MIN Datapath



Instruction Decoders

- Two decoders
- First Decoder (IB decoder)
  - Points to the first control word in an address mode sequence (if there is one)
  - The last state in any execution sequence is IB
- Second Decoder (SB decoder)
  - Points to the first control word of the execution sequence
  - The last sequence in addr. mode seq. is SB

Instruction Execution Sequences

Instruction	Control Word Sequence	Next control word address	IB Instruction Decoder	SB Instruction Decoder
POP	popr1	popr2	---	---
	popr2	brzz3	---	---
	brzz3	brzz2	---	---
	brzz2	---	abdm1	oprm1
ADD RX(RY +d)@	abdm1	abdm2	---	oprm1
	abdm2	abdm3	---	oprm1
	abdm3	abdm4	---	oprm1
	abdm4	---	---	oprm1
	oprm1	oprm2	---	---
	oprm2	brzz3	---	---
	brzz3	brzz2	---	---
	brzz2	---	oprr1	---

Instruction Execution Sequences

Instruction	Control Word Sequence	Next control word address	IB Instruction Decoder	SB Instruction Decoder
SUB RX RY	oprr1 oprr2 brzz2	oprr2 brzz2 ---	--- --- adrm1	--- --- test1
TEST RY@	adrm1 test1 ldrm2	--- ldrm2 ---	adrm1 --- push1	test1 --- ---
PUSH				

IB Instruction Decoder

IB Decoder Address	Instruction(s) or Address Mode	
abdm1	(RY+d)@	Address mode sequences
adrm1	RY@	
brzz1	BZ	Execution sequences (Instructions without separate address mode sequences)
ldrr1	LR	
strr1	STR	
oprr1	AR, SR, NR	
popr1	POP	
push1	PUSH	

SB Instruction Decoder

SB Decoder Address	Instruction(s) or Address Mode	
ldrm1	L	Execution sequences (Instructions with separate address mode sequences)
strm1	ST	
oprm1	A, S, N	
test1	T	

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

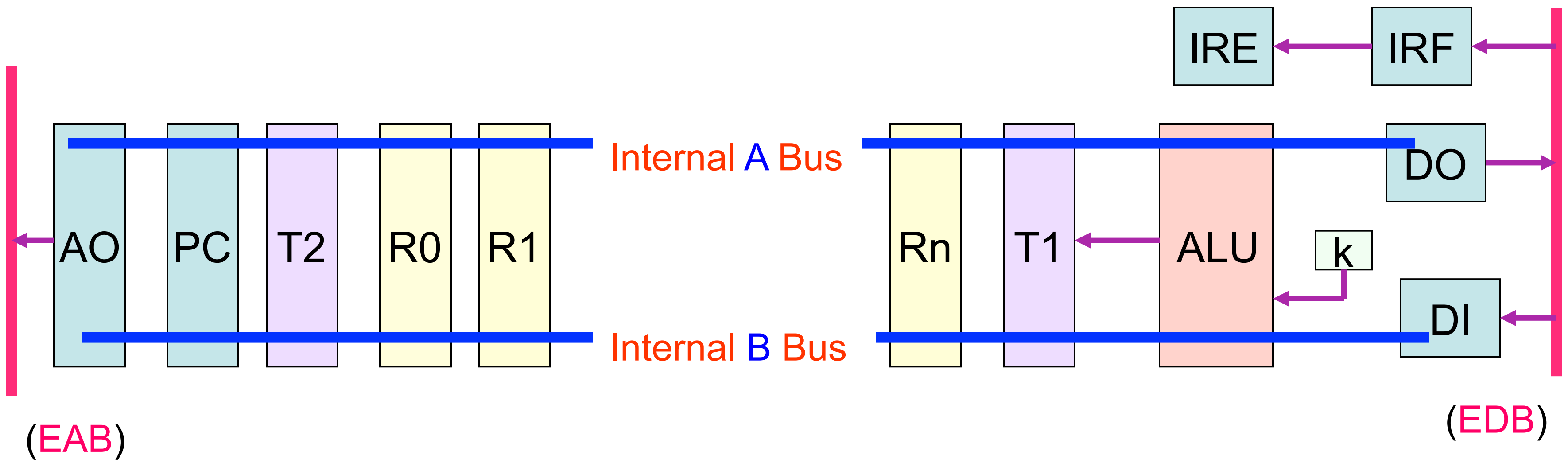
OP	TY	NA

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

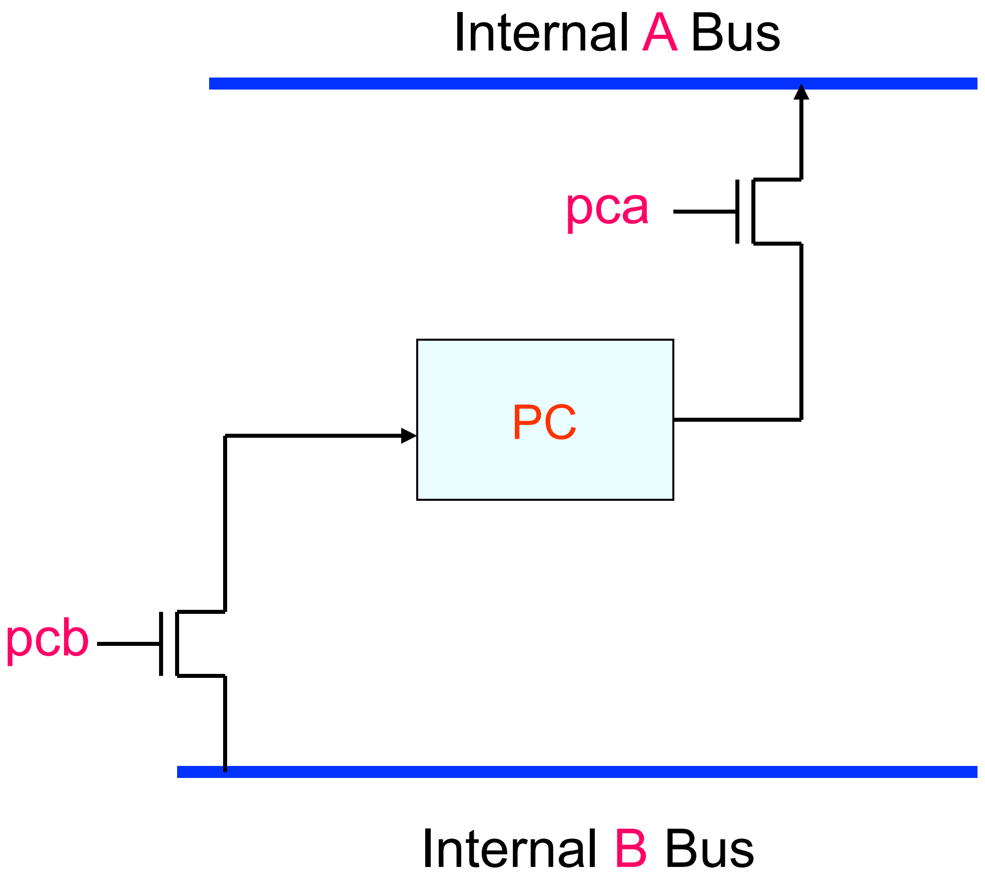
- Collect all the occurrences (PC, T2, RX …)
- Assign no. of bits to control fields

Control Word Decoder

- PC Control
  - PC occurrences
    - pc → a
    - a → pc (only one occurrence – abdm2)
    - b → pc
    - none

PC Control

pcb \ pca	0	1
0	none	pc → a
1	b → pc	x



Thank You