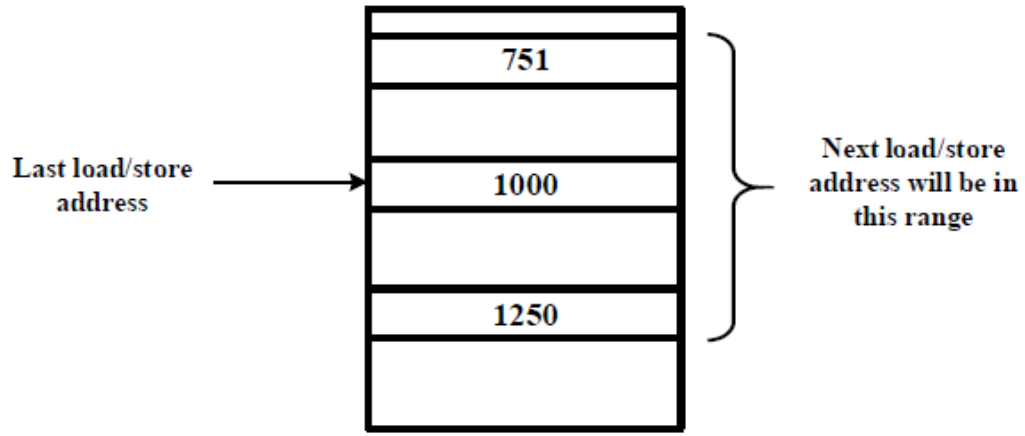
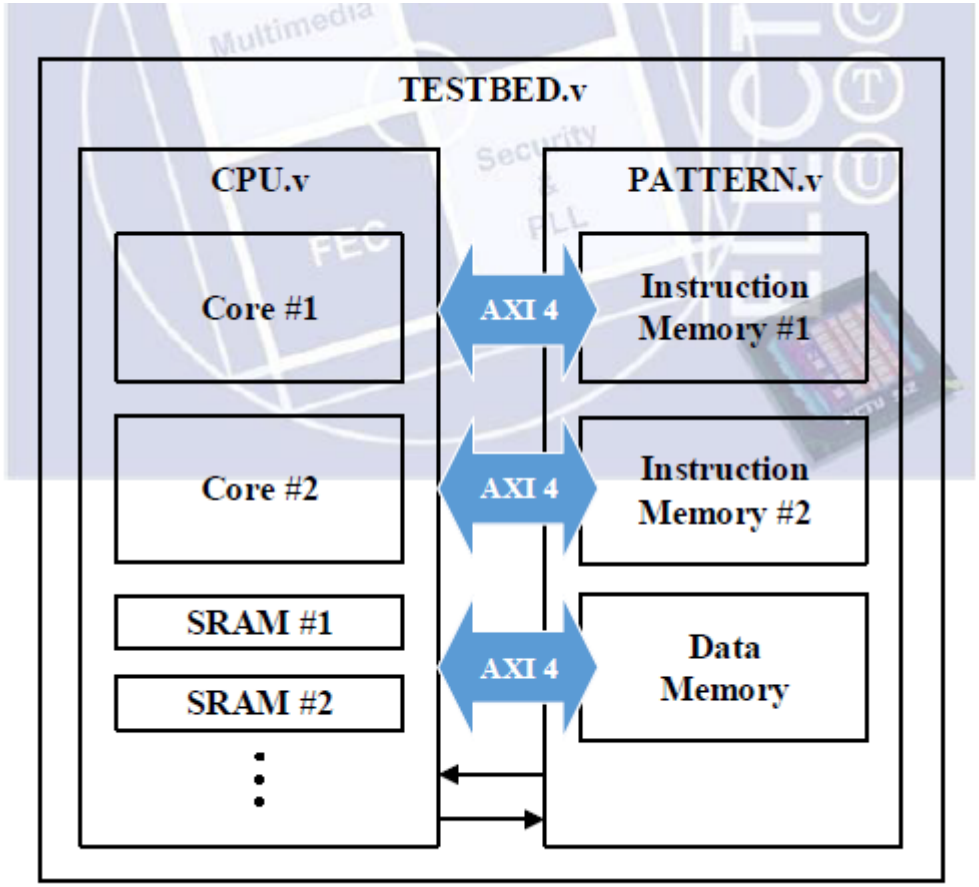
Customized ISA Processor

Zong-Ru Li

SPEC

- design two 16-bits CPU with 8 registers
- each core has one instruction memory but two core share one data memory
- the range of load/store, jump/branch is from (current address – 250 + 1) to (current address + 250)



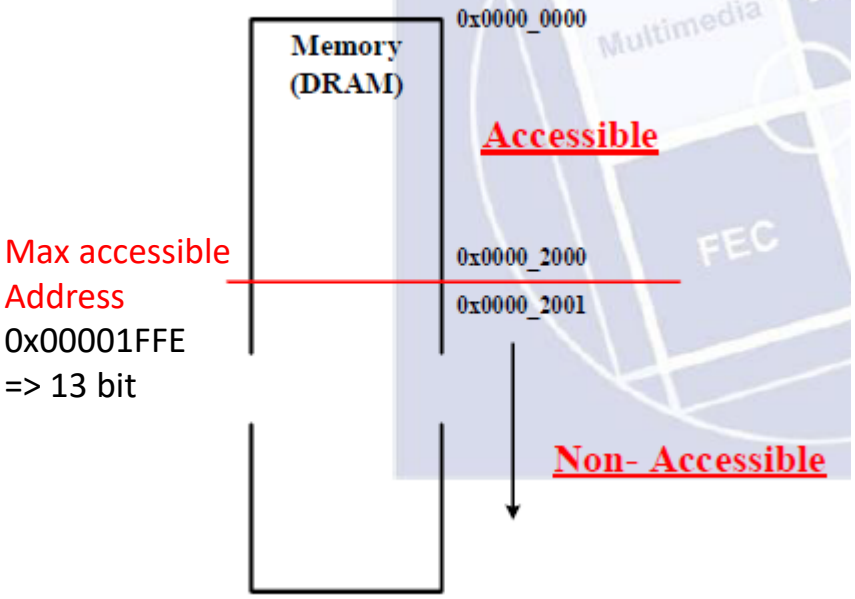


ISA

The following are the four formats for the core instruction set

Type	-15-	- Format – (16-bits)				-0-
R	opcode (3-bits)	rs (3-bits)	rt (3-bits)	rd (3-bits)	func (3-bits)	?
Mult	opcode (3-bits)	rs (3-bits)	rt (3-bits)	rd (3-bits)	rl (3-bits)	?
I	opcode (3-bits)	rs (3-bits)	rt (3-bits)	immediate (7-bits)		
J	opcode (3-bits)	Address (13-bits)				

- Memory available range



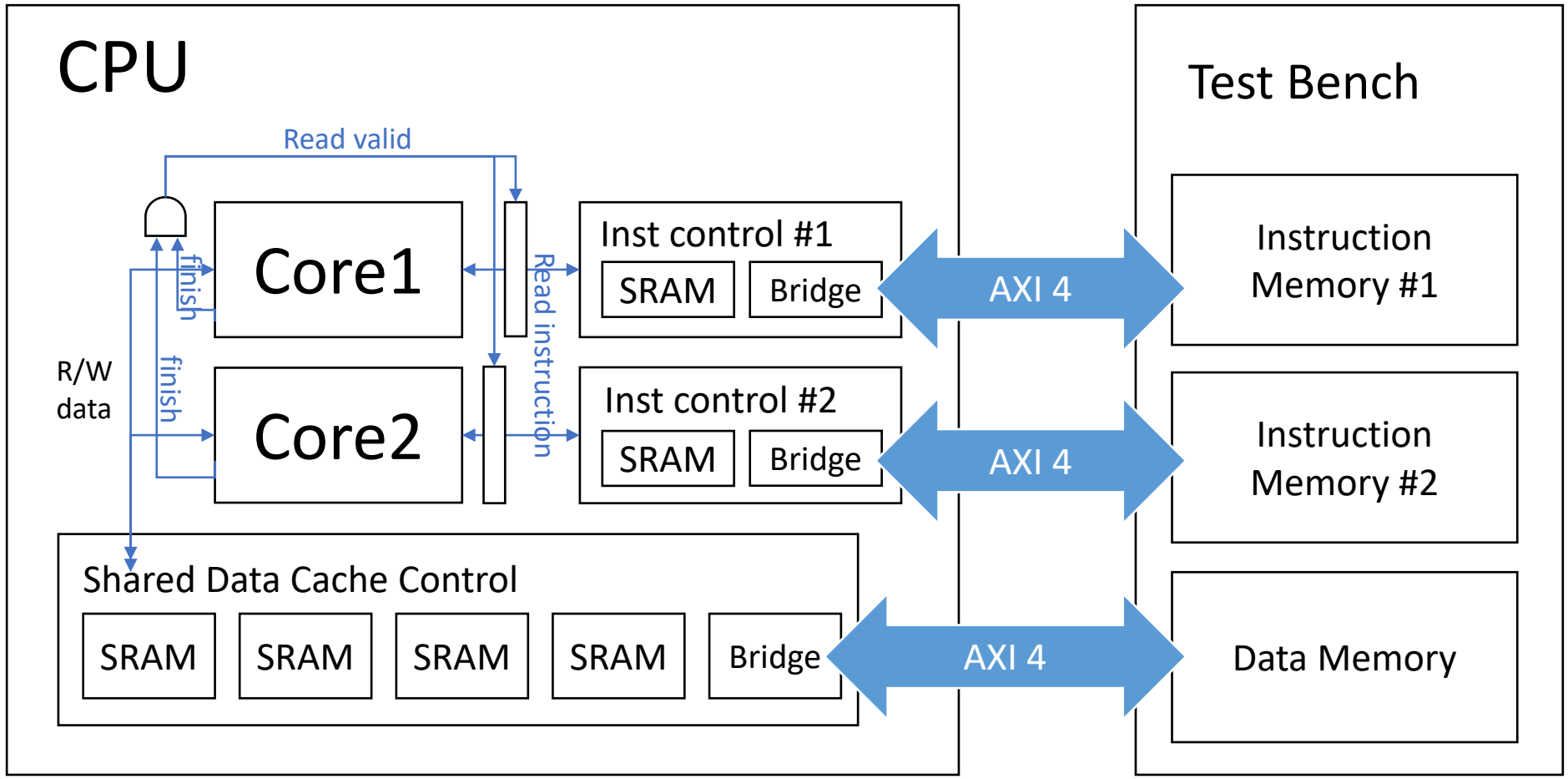
Detailed instructions are shown below

Function Name	Meaning	Type	Instruction Binary Encode
ADD	rd = rs + rt	R	000-rs-rt-rd-000?
SUB	rd = rs – rt	R	000-rs-rt-rd-001?
AND	rd = rs & rt (bit-wise)	R	000-rs-rt-rd-010?
OR	rd = rs | rt (bit-wise)	R	000-rs-rt-rd-011?
NAND	rd = ~(rs & rt) (bit-wise)	R	000-rs-rt-rd-100?
NOR	rd = ~(rs | rt) (bit-wise)	R	000-rs-rt-rd-101?
XOR	rd = rs ^ rt (bit-wise)	R	000-rs-rt-rd-110?
Set less than	if(rs<rt) rd=1 else rd=0	R	000-rs-rt-rd-111?
Mult	{rd, rl} = rs * rt	Mult	001-rs-rt-rd-rl?
ADDI	rt = rs+ immediate (sign)	I	010-rs-rt-iiiiiii
SUBI	rt = rs – immediate (sign)	I	011-rs-rt-iiiiiii
Load	rt = DM[rs+immediate(sign)]	I	100-rs-rt-iiiiiii
Store	DM[rs+immediate(sign)] = rt	I	101-rs-rt-iiiiiii
Branch on equal	if(rs==rt) pc=pc+1+immediate(sign)	I	110-rs-rt-iiiiiii
Jump	pc = address	J	111-address

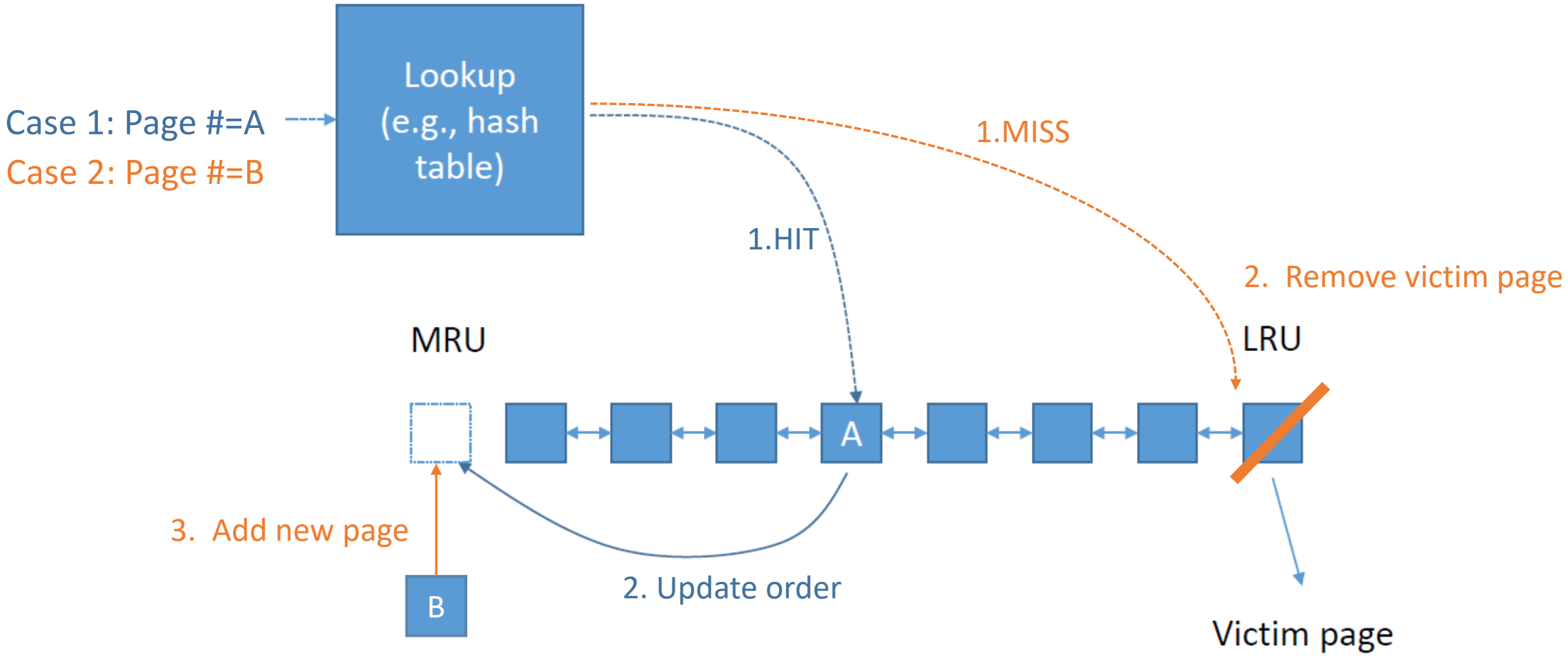
*pc : program counter

*? : useless bit

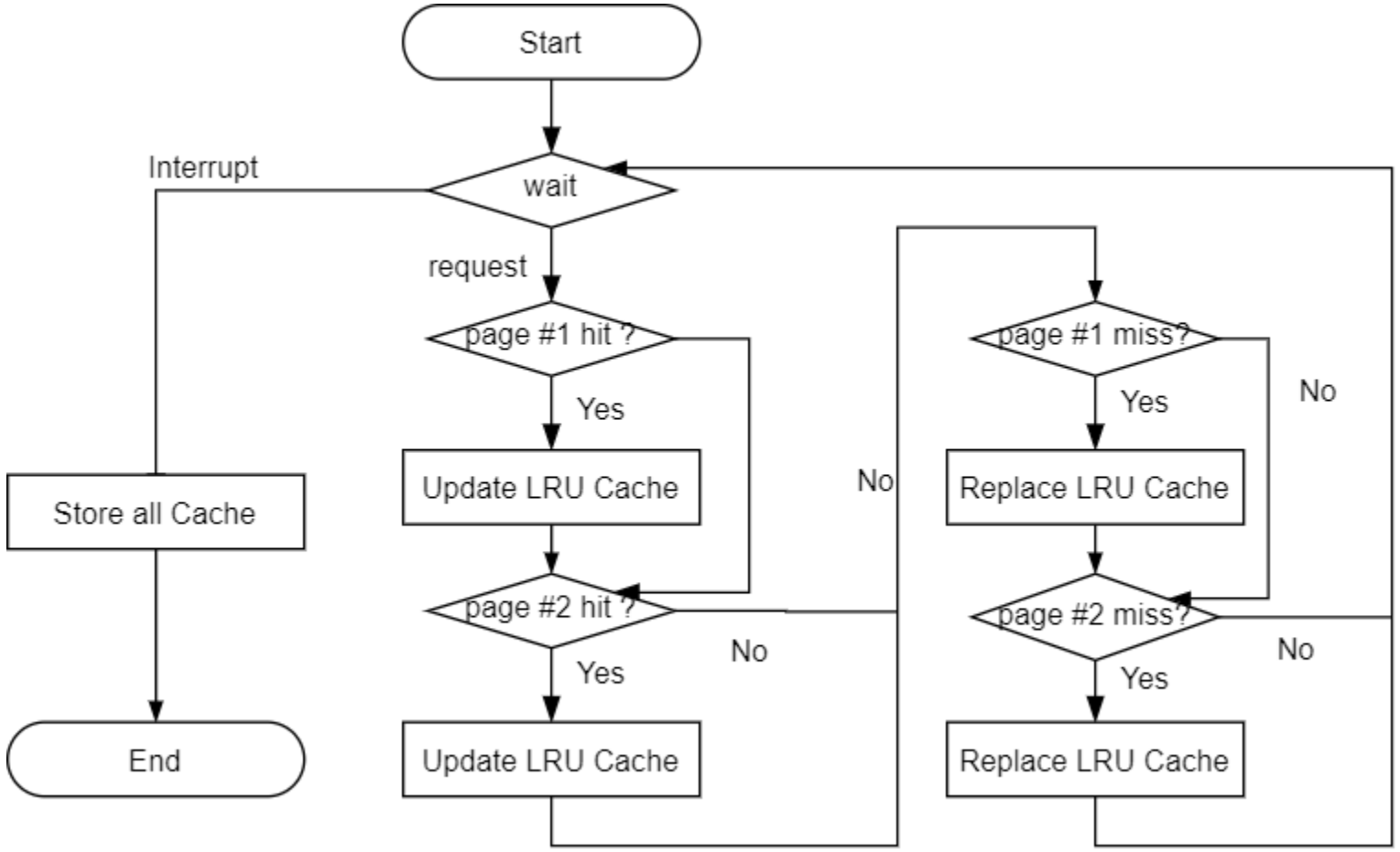
CPU TOP LEVEL



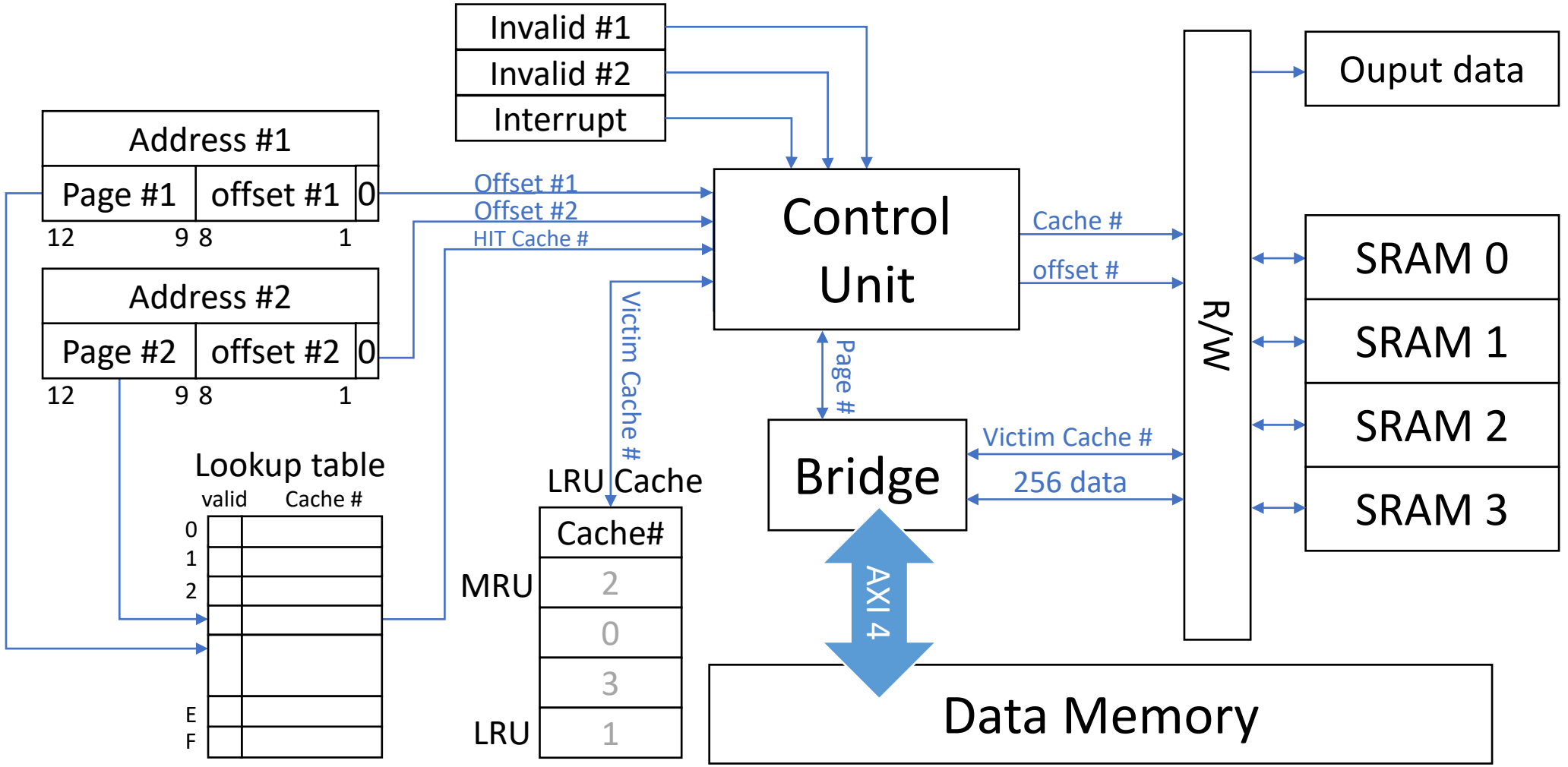
LRU simulator structure



Shared Data Cache control flow chart



Shared Data Cache architecture

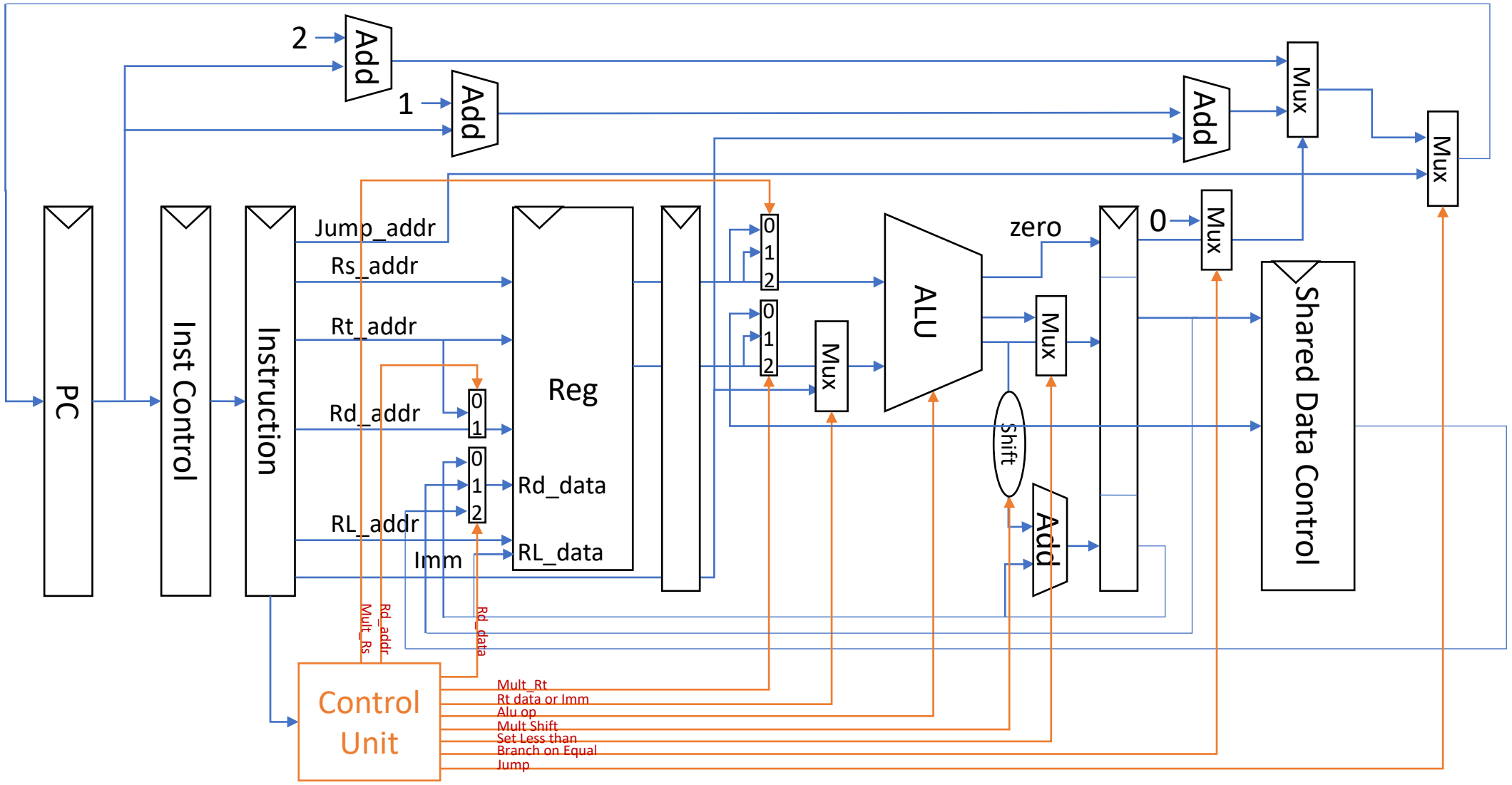


Penalty

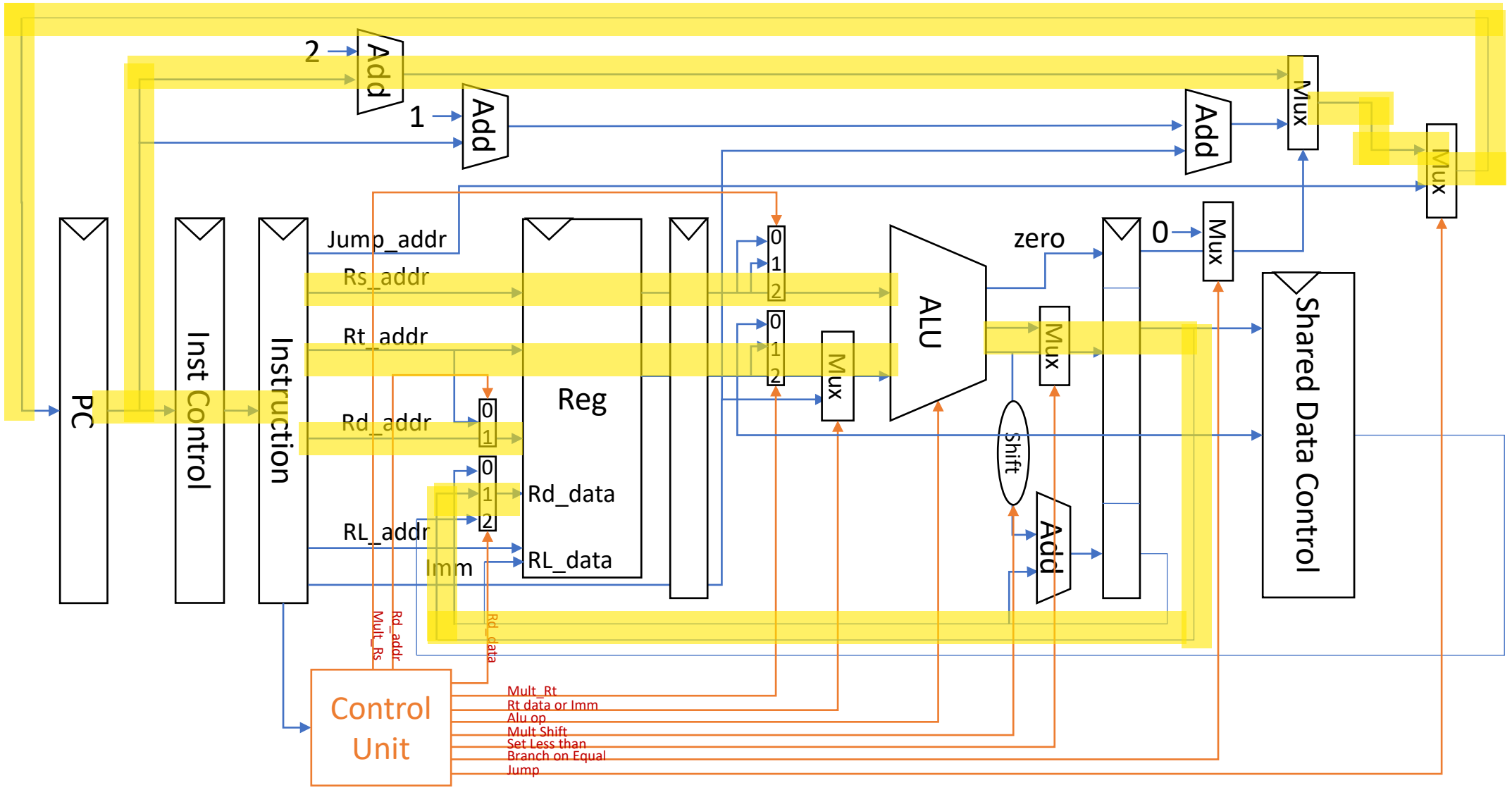
Execution time = 1000*CPU cycle+1000*access rate*(hit+ miss rate*miss penalty)

1	sram #	CPU cycle	access rate	hit	LRU miss rate	miss penalty	total execution time	speed up(%)
2	0	6	0.13	1	1	100	19130	
3	1	6	0.13	1	0.87	350	45715	
4	2	6	0.13	1	0.18	350	14320	25
5	4	6	0.13	1	0.079	350	9724.5	49

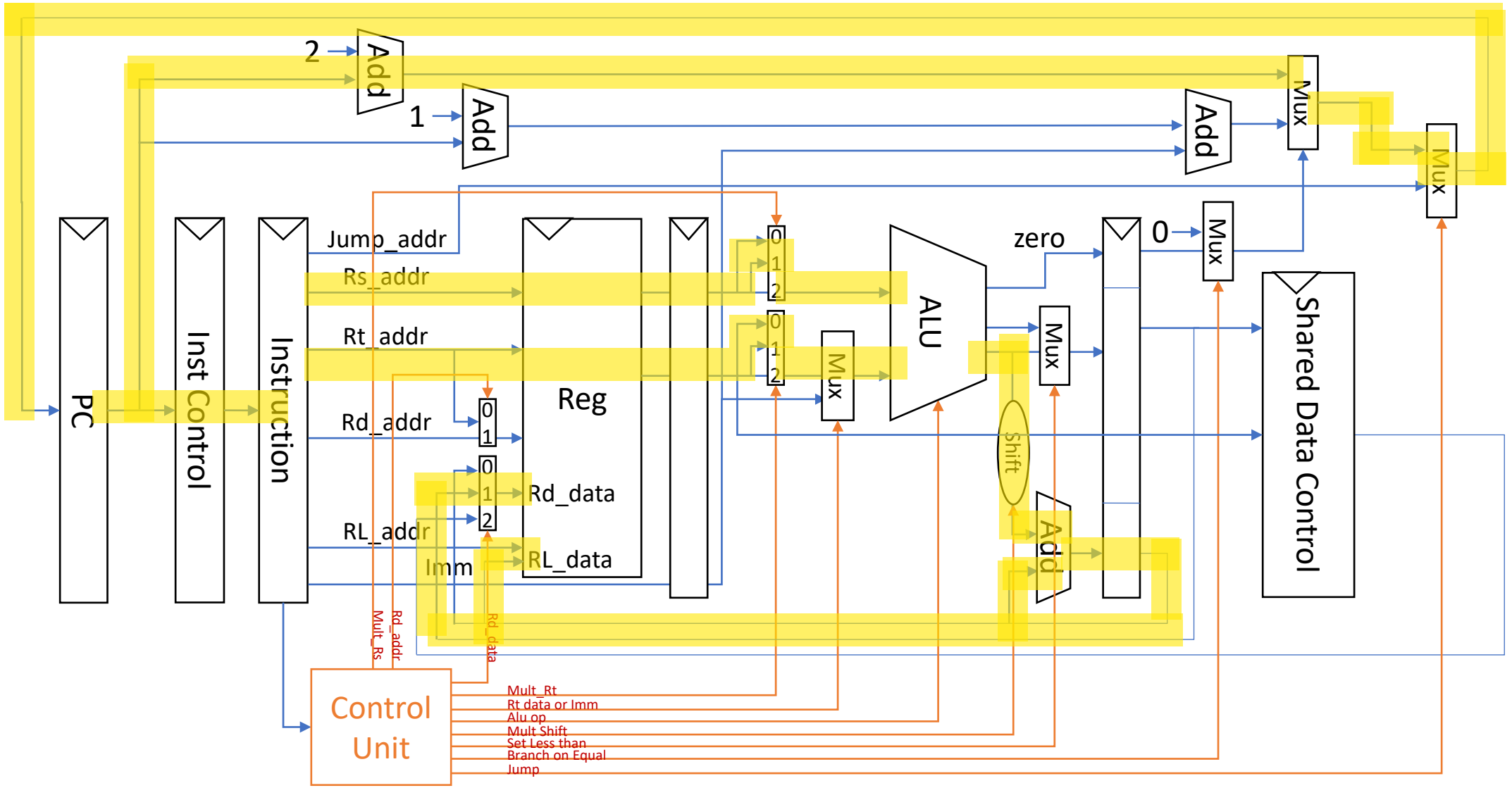
Single Core architecture



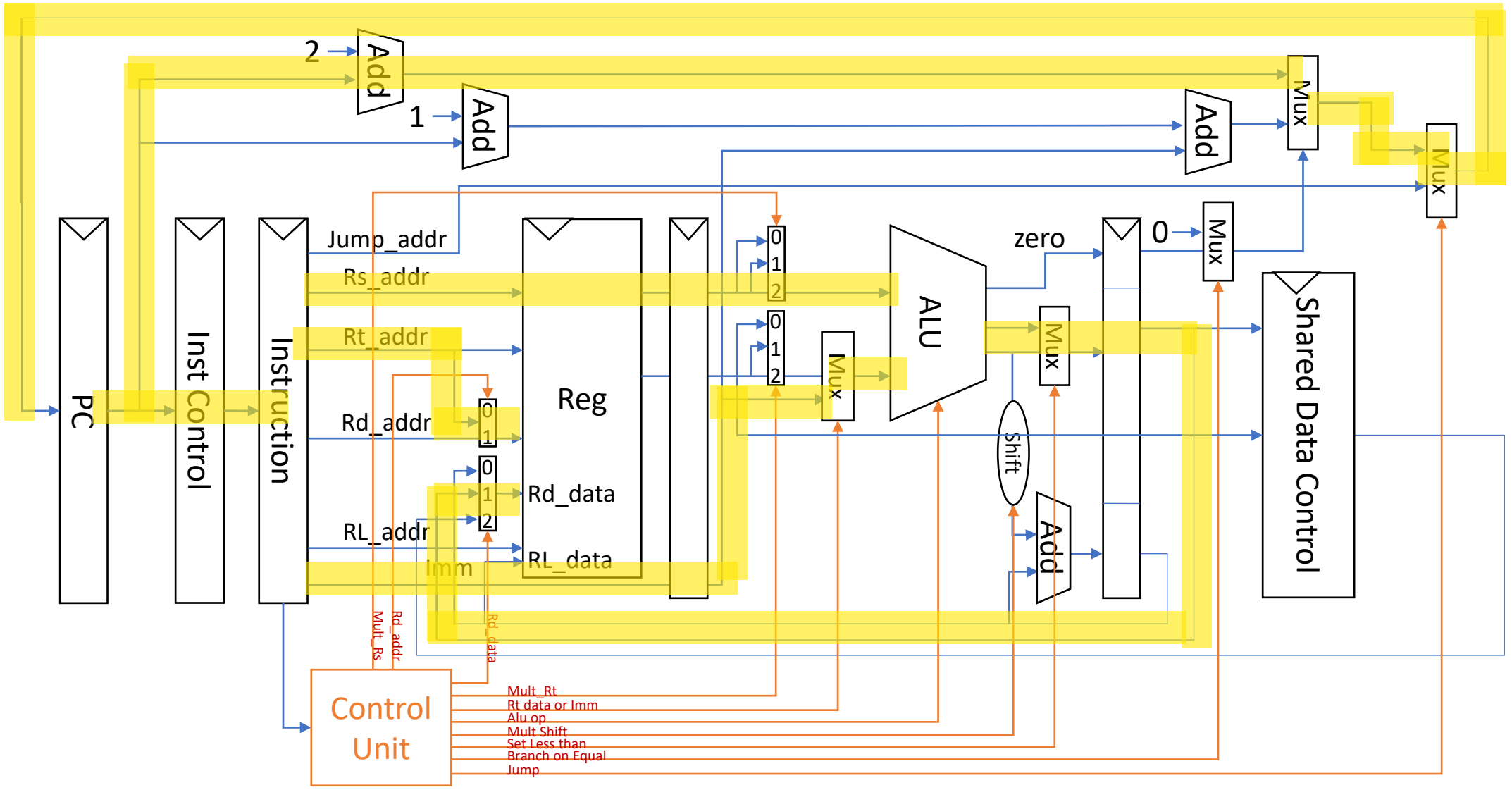
R-type data flow



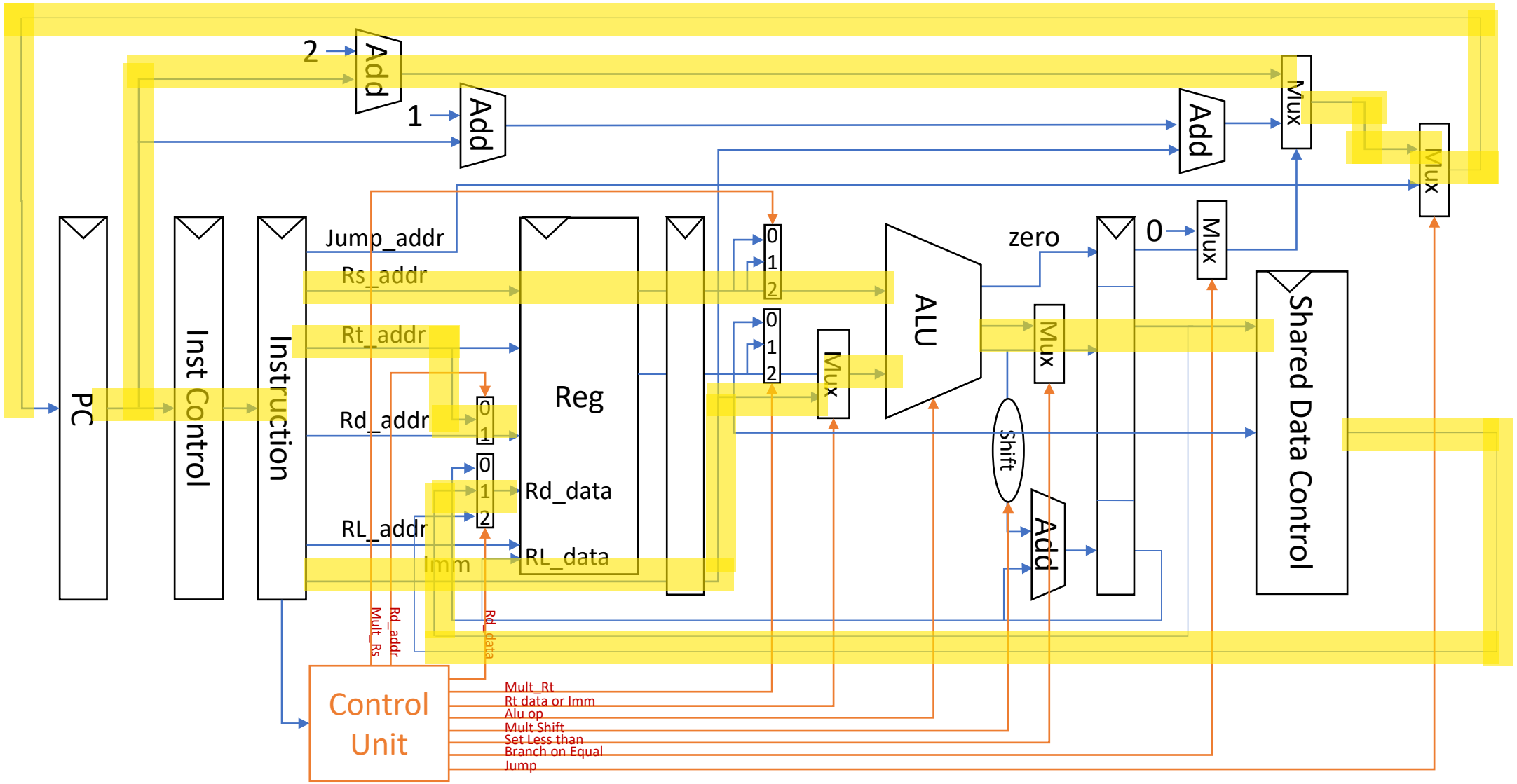
Mult data flow



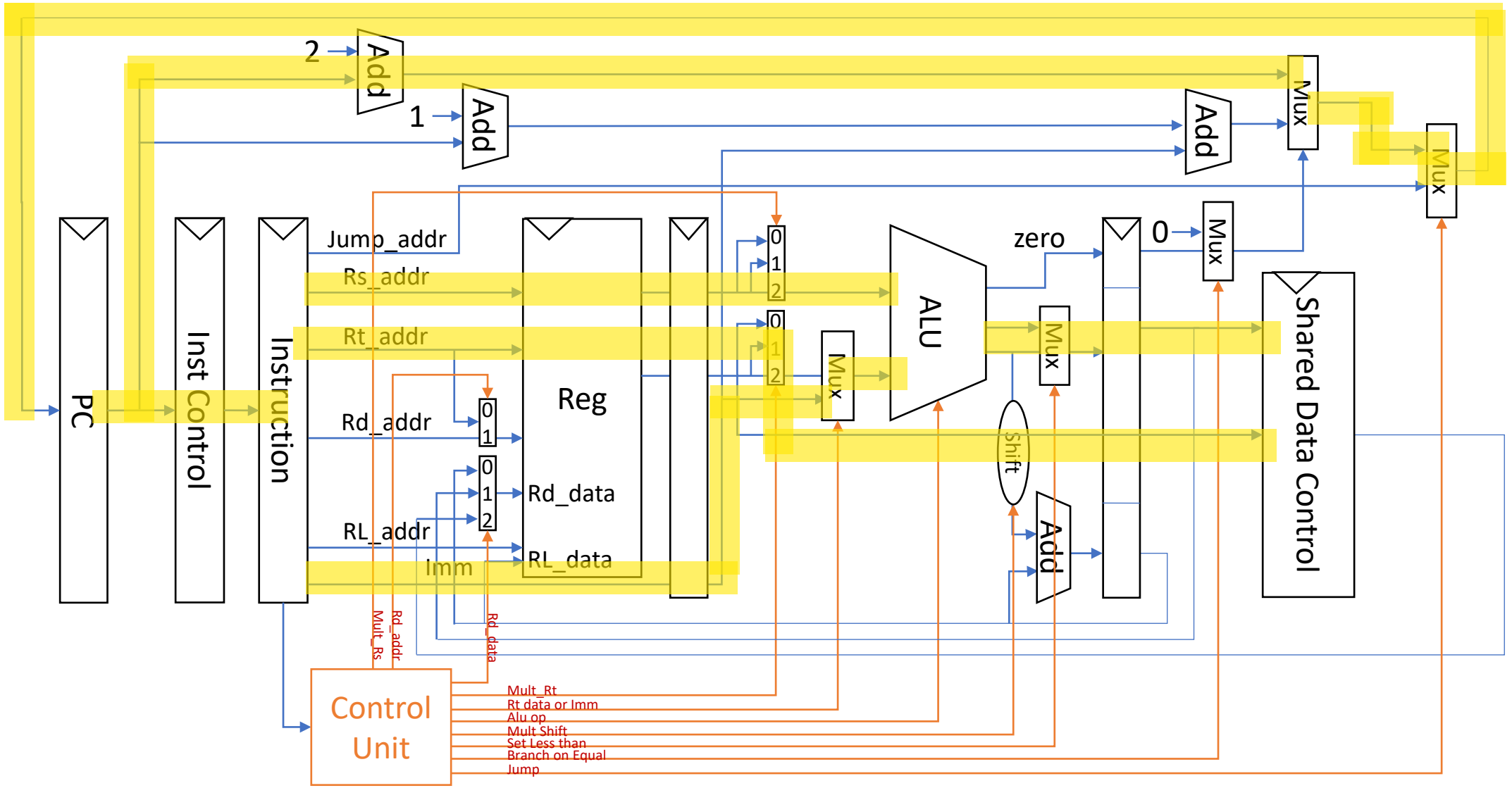
I-type data flow



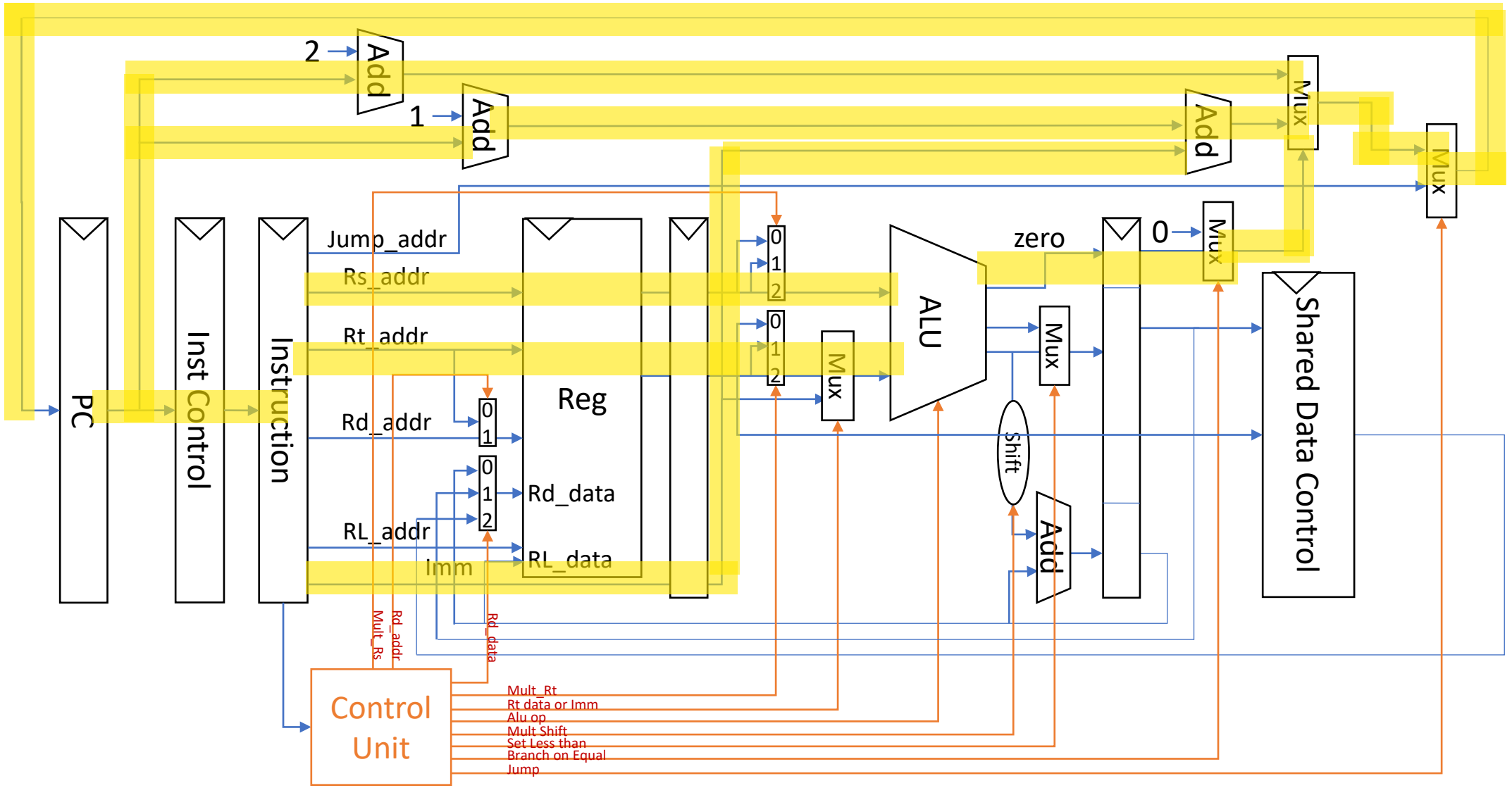
Load data flow



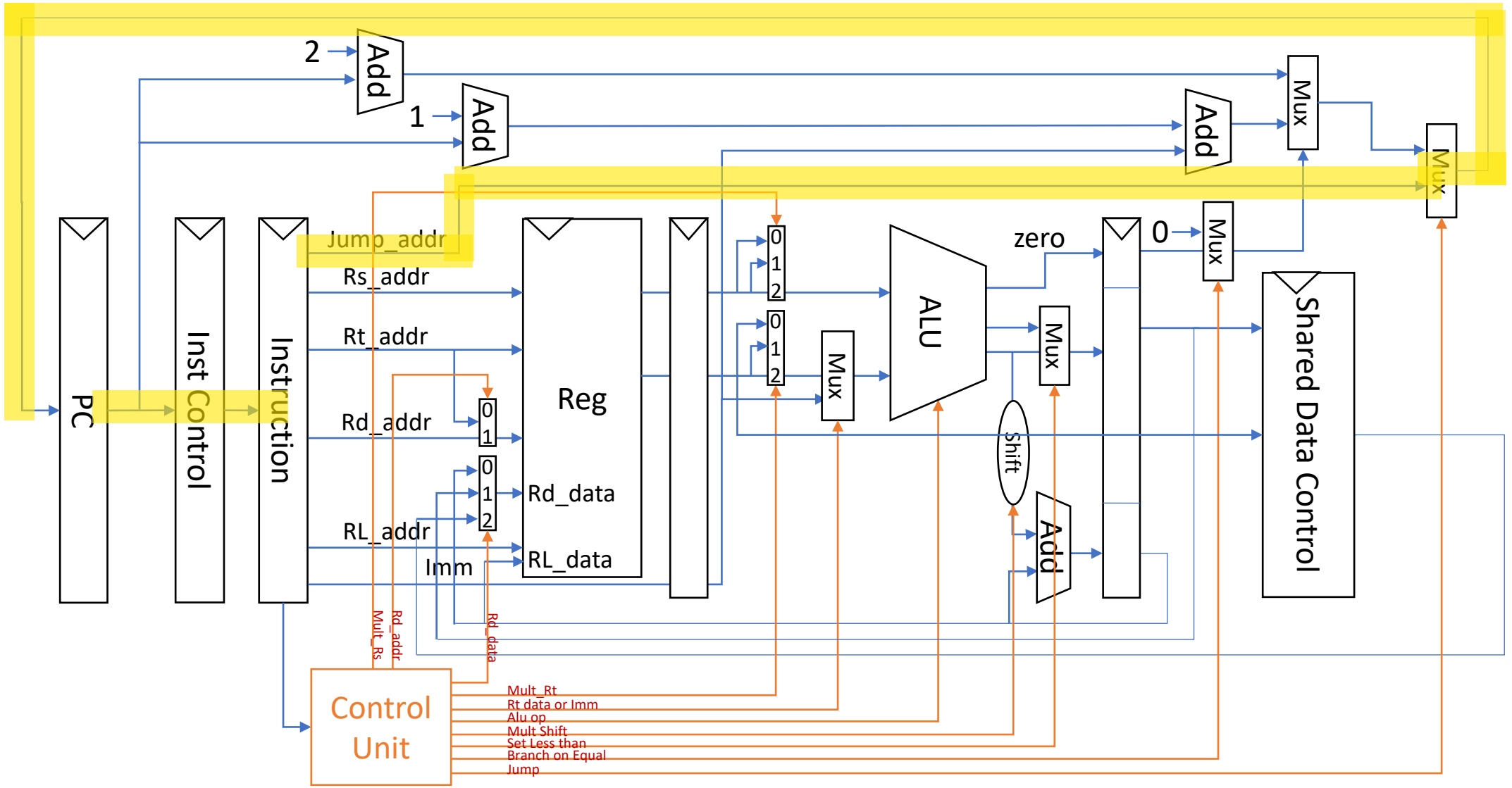
Store data flow



Branch on Equal data flow



J-type data flow



Thanks for listening