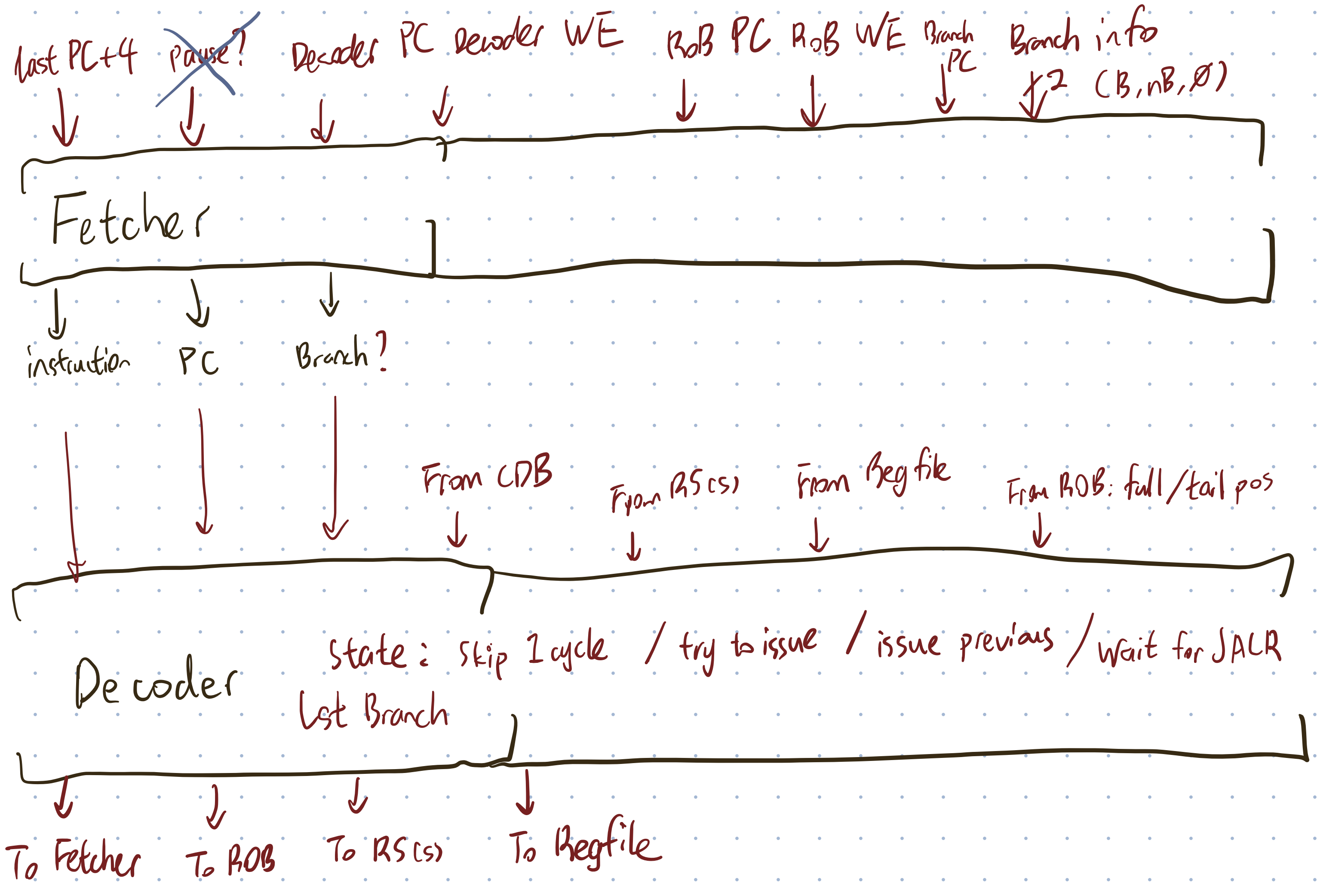


Branch: if $Branch? is true, state ⇐ SKIP; Calculate the new PC write to Fetcher. and issue.

otherwise, issue.

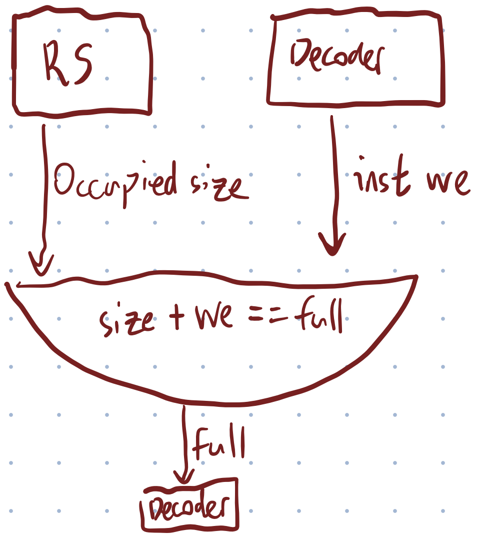
JAL: Calculate the new PC. write to Fetcher, write to ROB

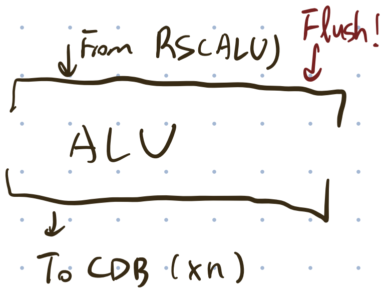
Issue failure: write PC+4 to Fetcher, go to "issue previous" state

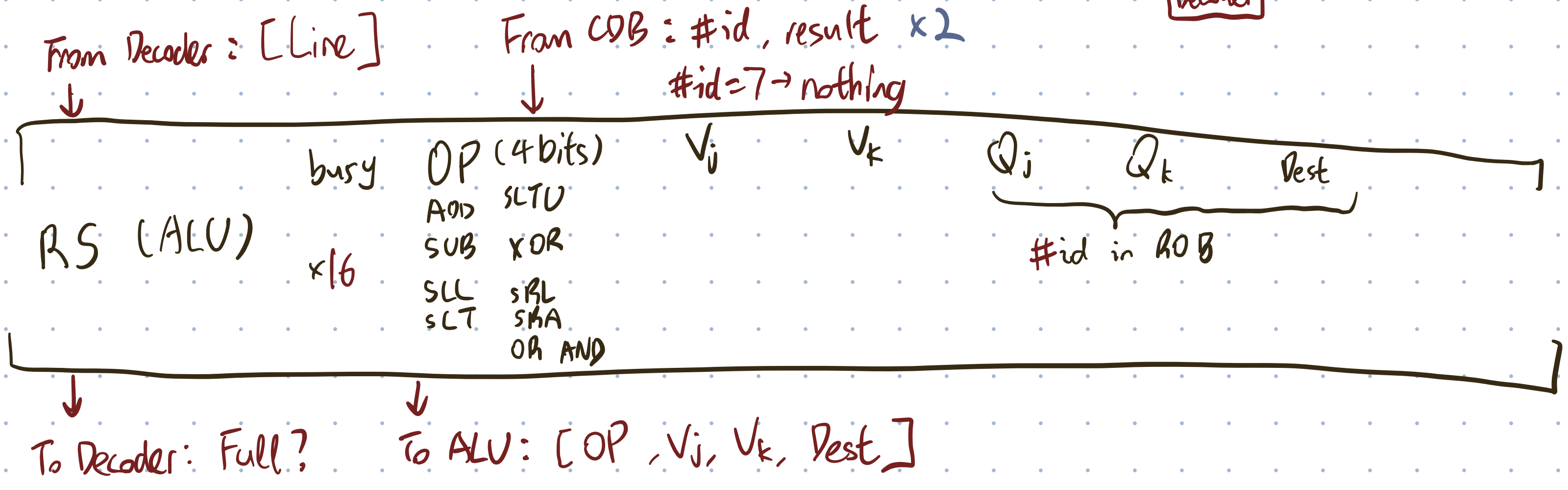
JALR: View as ADD, go to "Wait for JALR state"

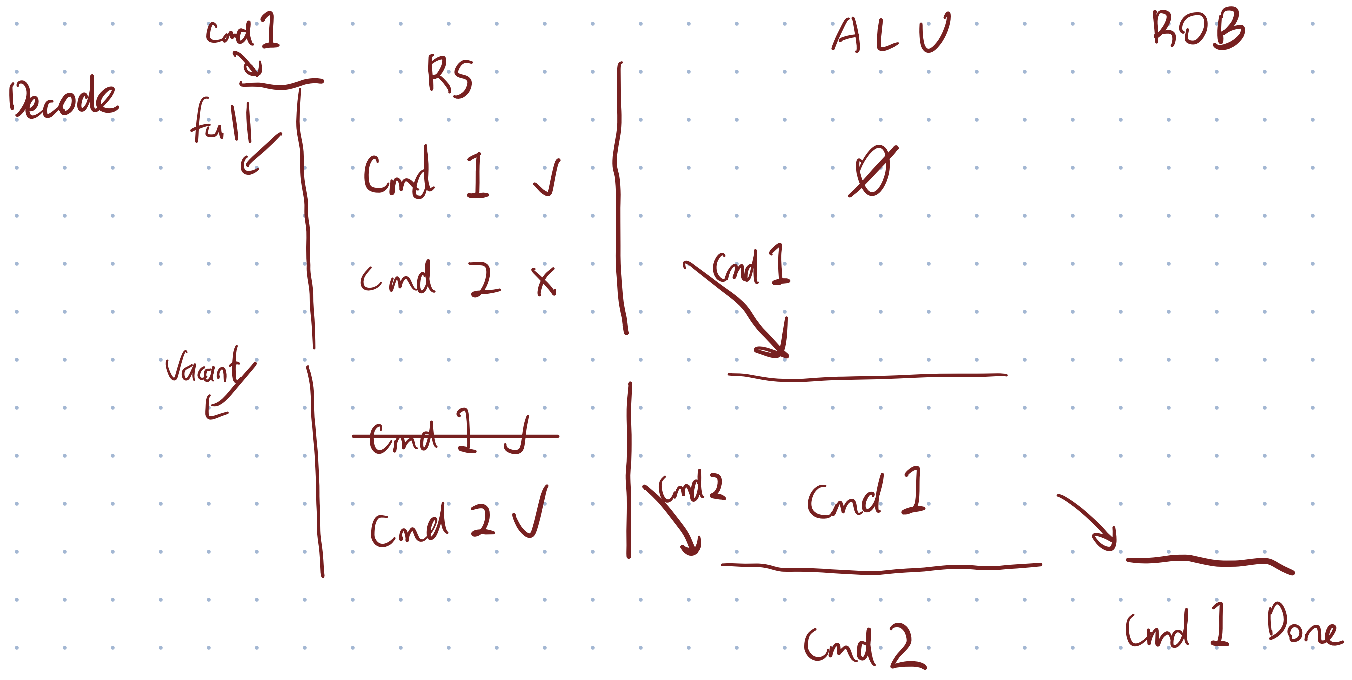
↳ Special Case: Ret, and x1 is ready → View as J, go to "skip 1 cycle" state

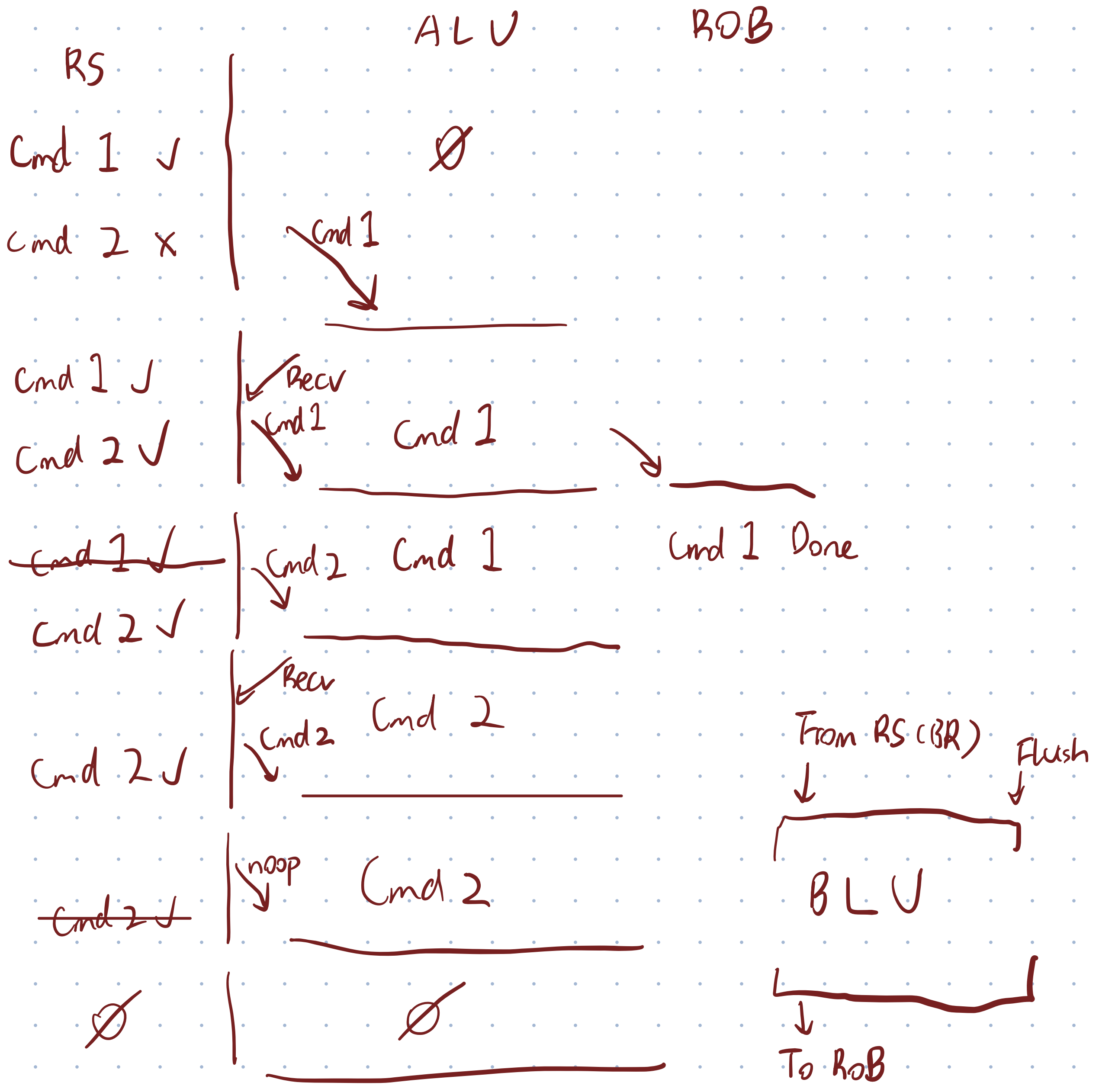
Fetch		Decode
x	↘Cmd 1	Skip
x+4	↘Cmd 2	Cmd 1 ✓
x+8	↘Cmd3	Cmd 2 ✓
x+12	↘Cmd 4 pause↙	Cmd 3 ✗
x+12	↘Cmd 4	Cmd 3 ✓
x+16	↘B1nB	Cmd 4 ✓
x+20	↘B2,B	B1 ✓
x+24	↘Sth PC=y↙	B2 ✓
y	↘Cmd5	Skip

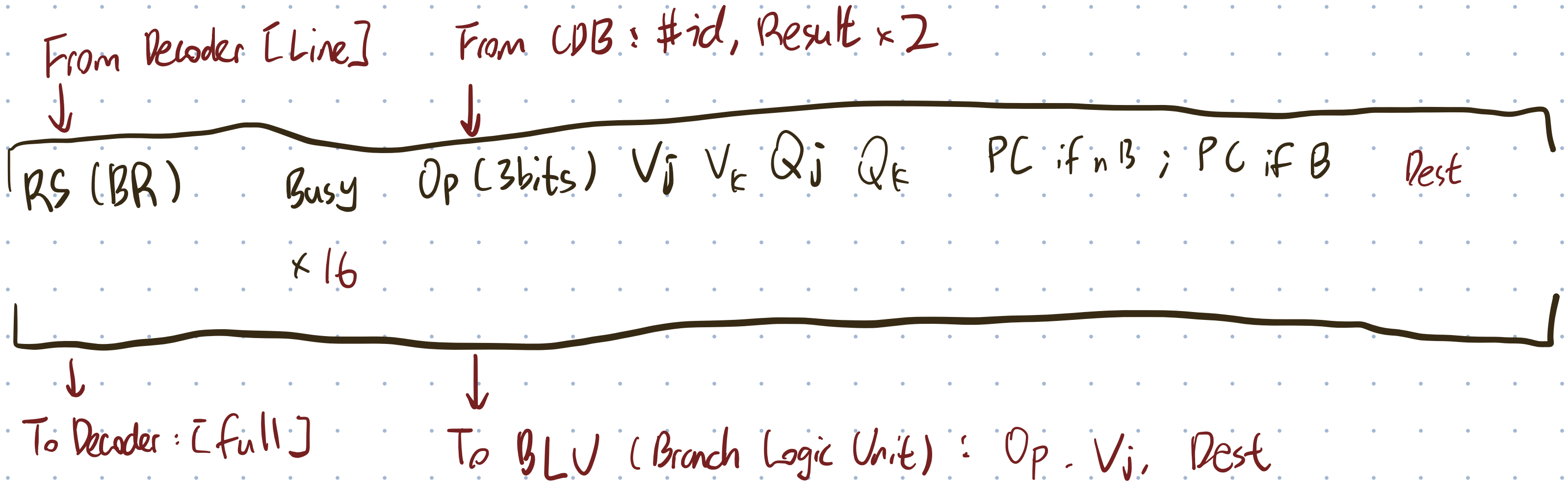


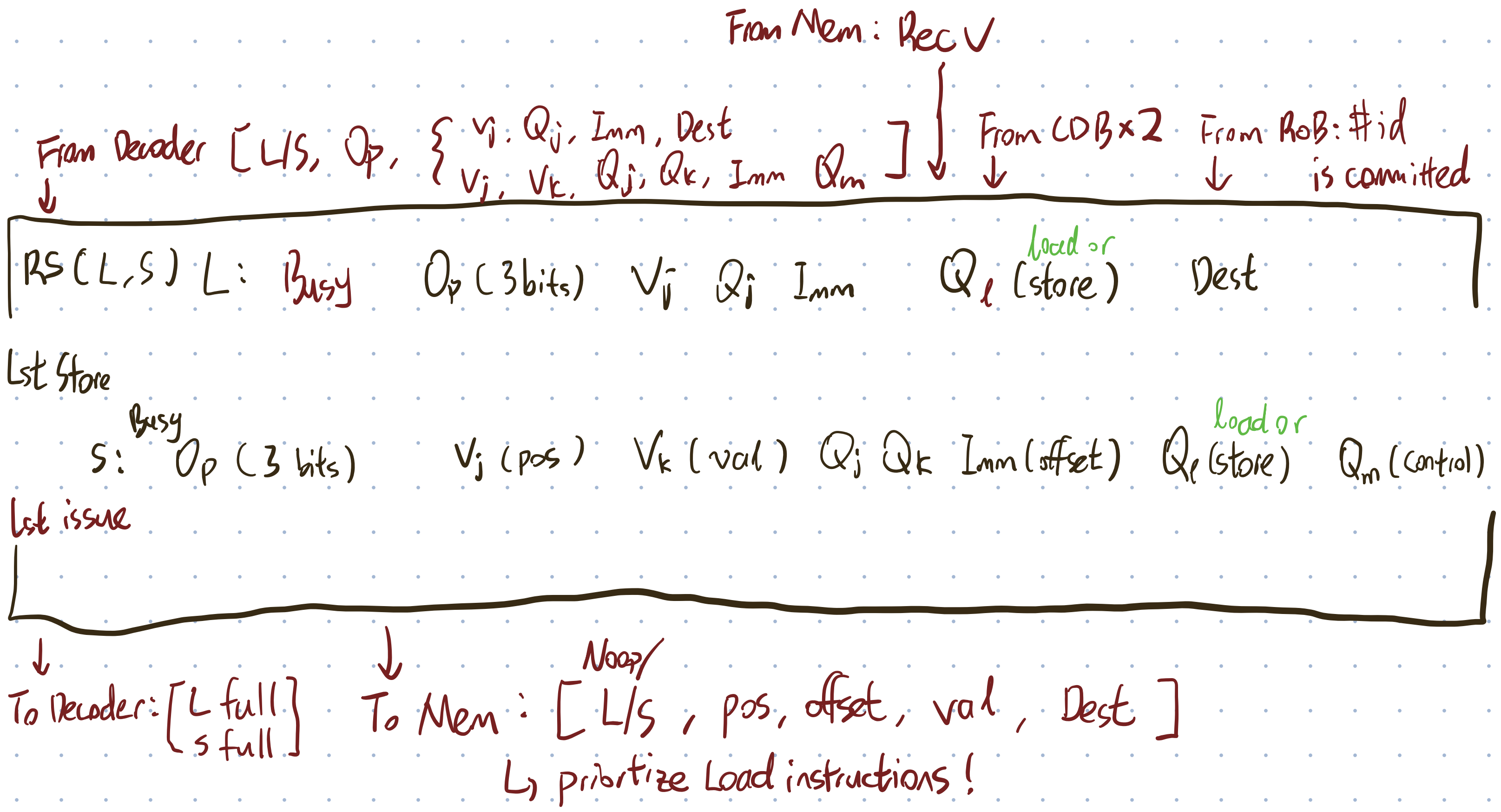




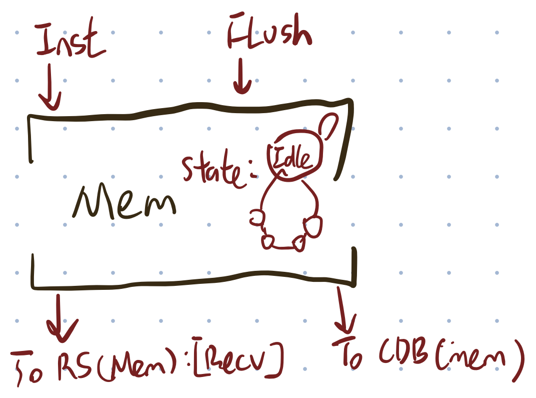

Decode
Cmd 1
full
Vacant
RS
Cmd 1 ✓
Cmd 2 ✗
~~Cmd 1 ✓~~
Cmd 2 ✓
ALU
∅
Cmd 1
Cmd 2
Cmd 1
Cmd 2
ROB
Cmd 1 Done

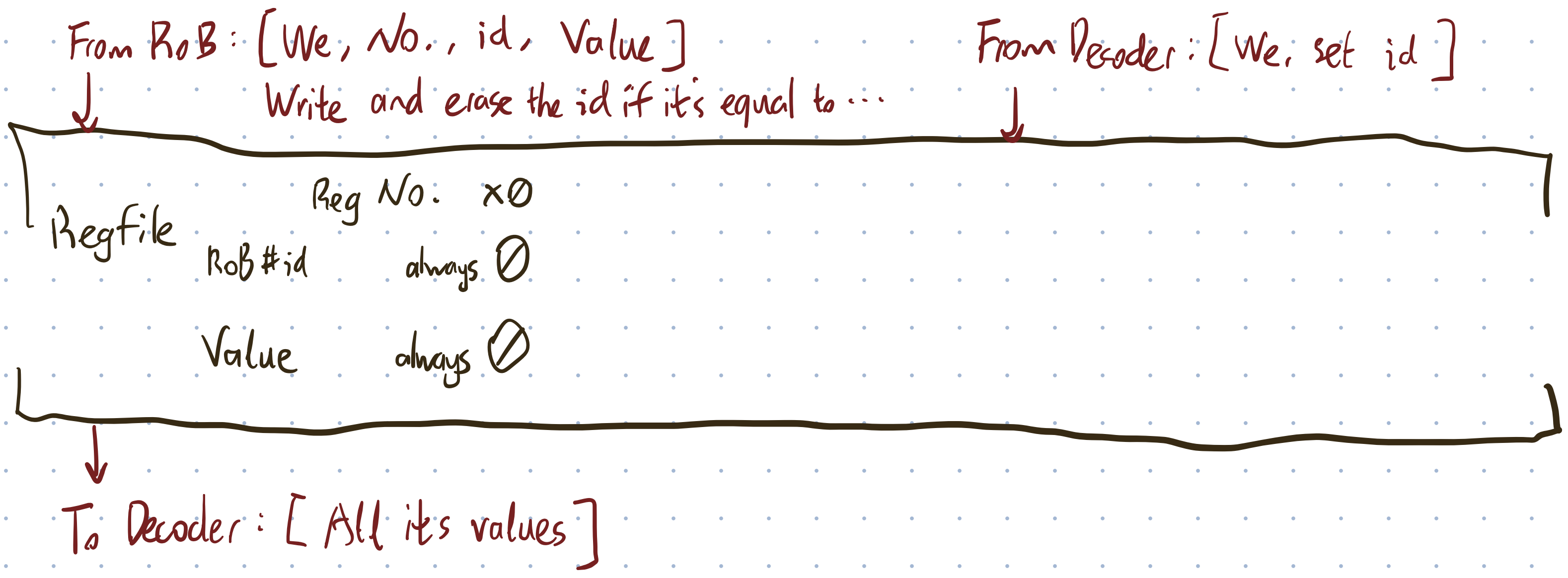

RS
Cmd 1 ✓
Cmd 2 ✗
Cmd 1 ✓
Cmd 2 ✓
~~Cmd 1 ✓~~
Cmd 2 ✓
Cmd 2 ✓
~~Cmd 2 ✓~~
∅
ALU
∅
Cmd 1
Recv
Cmd 1
Cmd 1
Cmd 2
Cmd 1
Recv
Cmd 2
Cmd 2
noop
Cmd 2
∅
ROB
Cmd 1 Done
From RS (BR)
Flush
BLU
To RoB

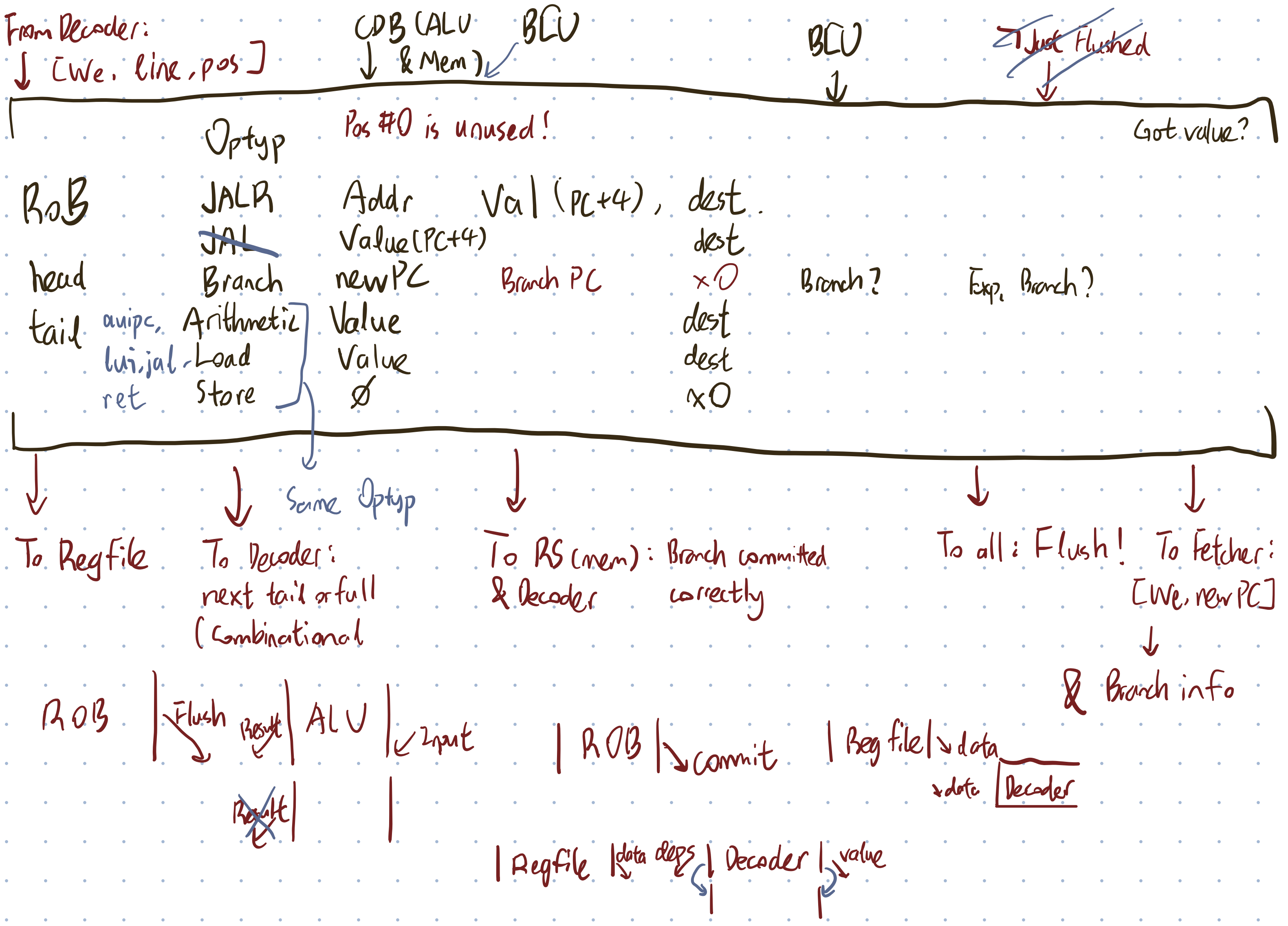



From Mem: Recv
From Decoder [L/S, Op, { Vj, Qj, Imm, Dest
Vj, Vk, Qj, Qk, Imm Qm]
From CDB×2
From RoB: #id is commited
RS(L,S) L: Busy Op (3bits) Vj Qj Imm Ql (store) Dest
load or
Lst Store
S: Busy Op (3 bits) Vj (pos) Vk (val) Qj Qk Imm (offset) Ql (store) Qm (control)
load or
Lst issue
To Decoder: [L full
S full]
To Mem: [Noop/ L/S, pos, offset, val, Dest]
↳ prioritize Load instructions!

If Recv is received, remove Ql that equals Lst issue.




From RoB: [We, No., id, Value]
Write and erase the id if it's equal to ...
From Decoder: [We, set id]
Regfile
Reg No. x0
RoB #id always 0
Value always ∅
To Decoder: [All its values]


From Decoder:
[We, line, pos]
CDB (ALU & Mem)
BCU
BCU
~~[Just~~ Flushed
Pos #0 is unused!
Got value?
Optyp
RoB
JALR Addr Val (PC+4), dest.
~~JAL~~ Value (PC+4) dest
head
Branch newPC Branch PC x0 Branch? Exp. Branch?
tail
auipc, Arithmetic Value dest
lui, jal, Load Value dest
ret Store ∅ x0
Same Optyp
To Regfile
To Decoder: next tail or full (Combinational
To RS (mem) & Decoder: Branch commited correctly
To all: Flush!
To Fetcher: [We, newPC]
& Branch info
ROB
Flush
Result
ALU
Input
~~Result~~
ROB
commit
Reg file
data
data
Decoder
Regfile
data deps
Decoder
value