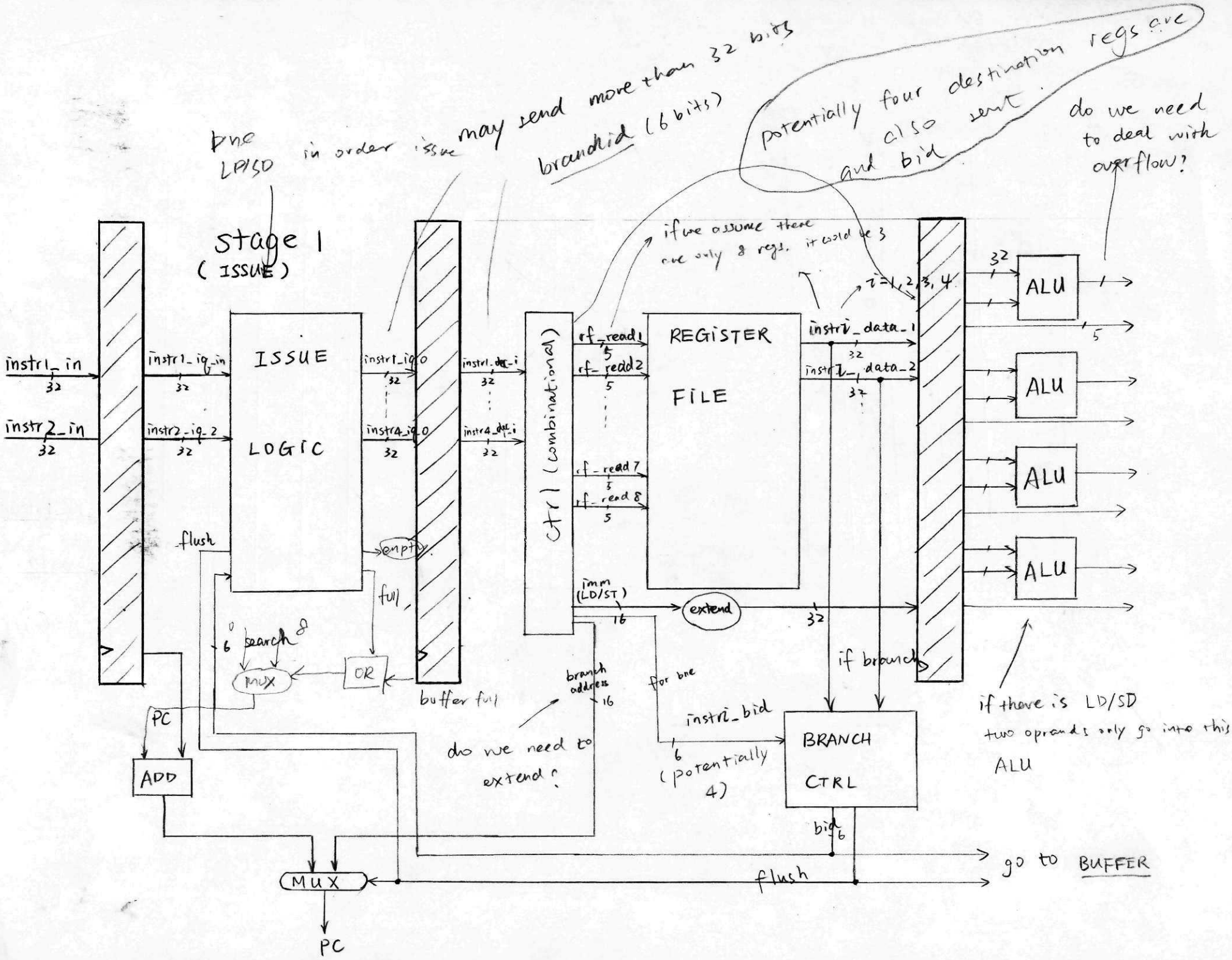
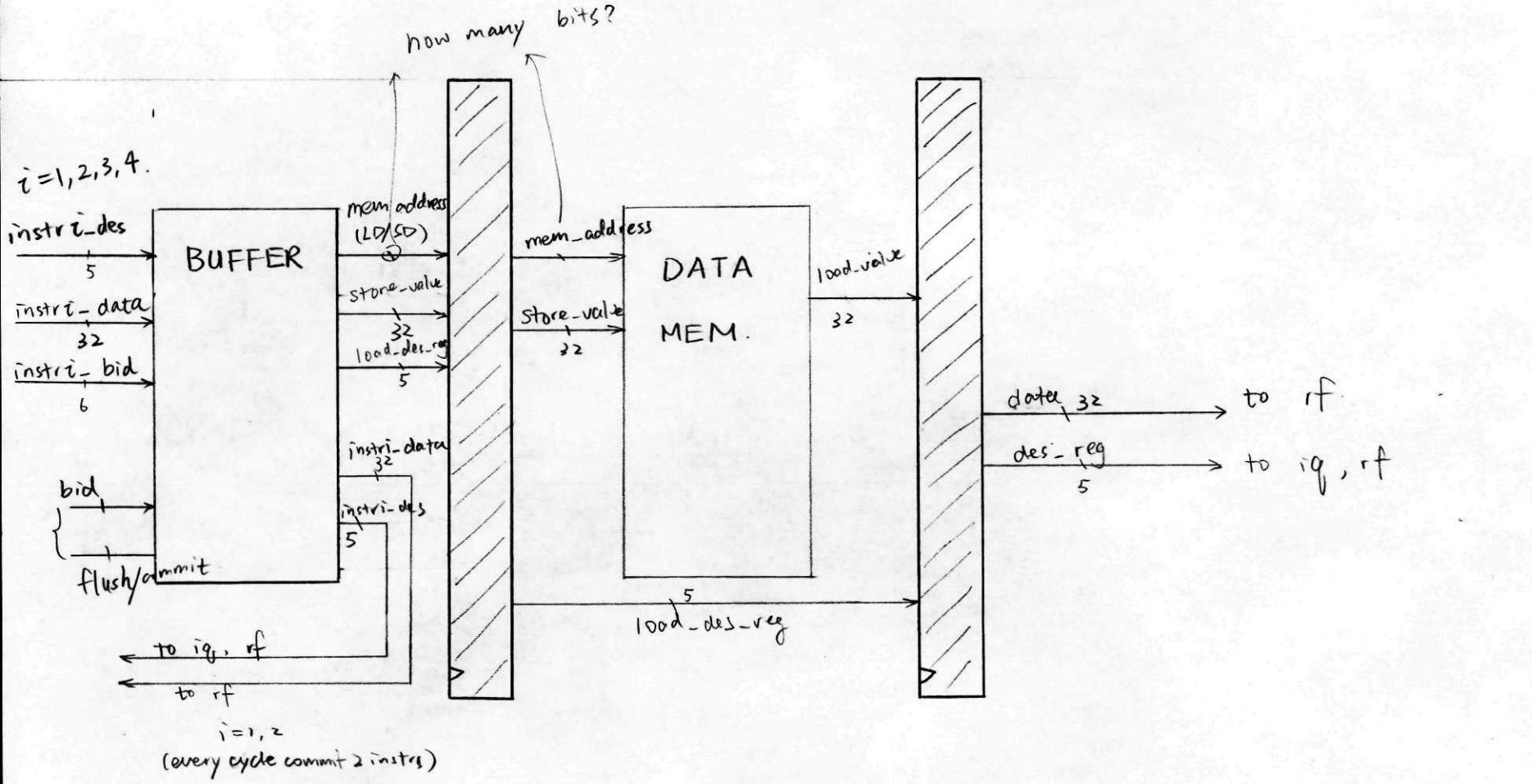

one LD/SD in order issue
may send more than 32 bits
brandhid (6 bits)
potentially four destination regs are also sent and bid
do we need to deal with overflow?
stage 1
(ISSUE)
if we assume there are only 8 regs, it could be 3
i = 1, 2, 3, 4
instr1_in
32
instr2_in
32
instr1_iq_in
32
instr2_iq_2
32
ISSUE
LOGIC
instr1_iq_0
32
instr4_iq_0
32
instr1_dst_i
32
instr4_dst_i
32
ctrl (combinational)
rf_read1
5
rf_read2
5
rf_read7
5
rf_read8
5
REGISTER
FILE
instri_data_1
32
instri_data_2
32
32
ALU
ALU
ALU
ALU
5
flush
empty
full
buffer full
6
search
8
MUX
OR
imm (LD/ST)
16
extend
32
if branch
branch address
16
for bne
instri_bid
6
(potentially 4)
BRANCH CTRL
bid
6
if there is LD/SD two oprands only go into this ALU
PC
ADD
do we need to extend?
MUX
PC
flush
go to BUFFER



For iq a buffer, full, empty

stall control?

PC

Stage 1 (ISSUE) (1)

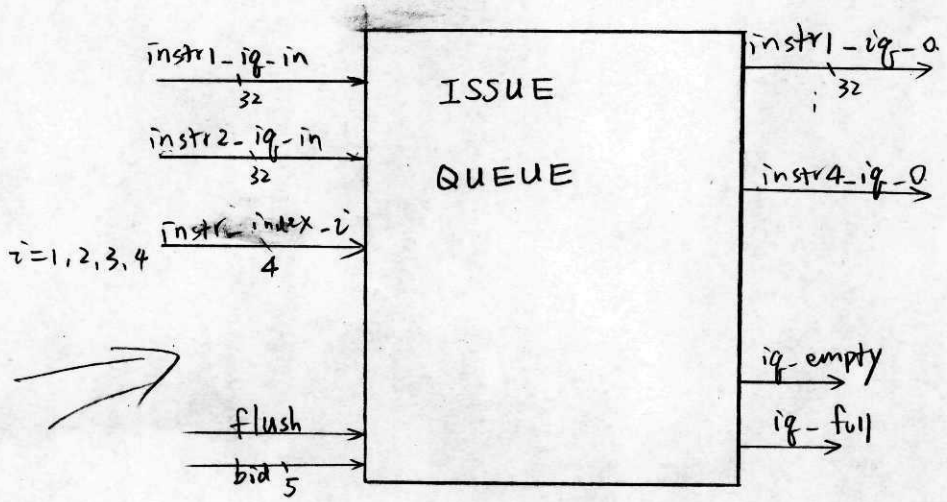
1. Basic Functionality

   ① Issue queue

   ② Hazard check

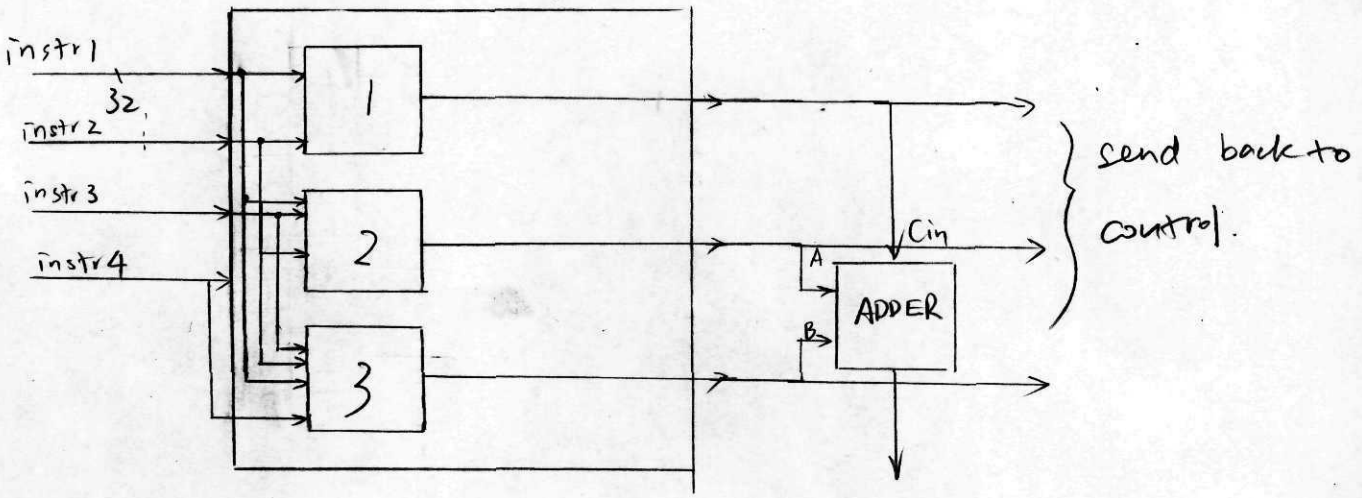
2. Basic Blocks

   a. Issue queue

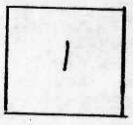


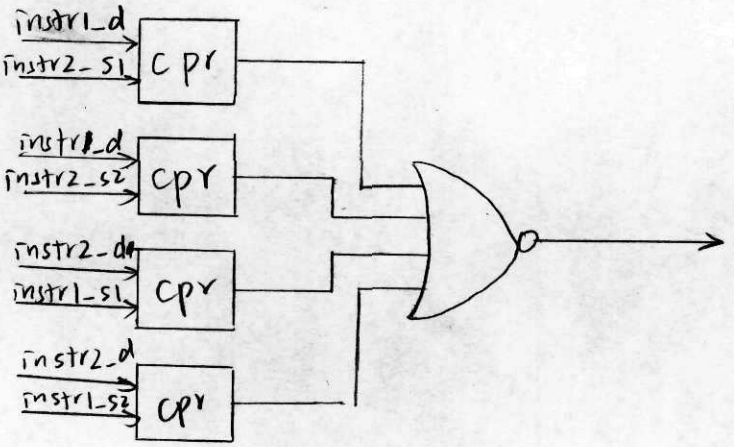
control signal

b. Hazard Checker



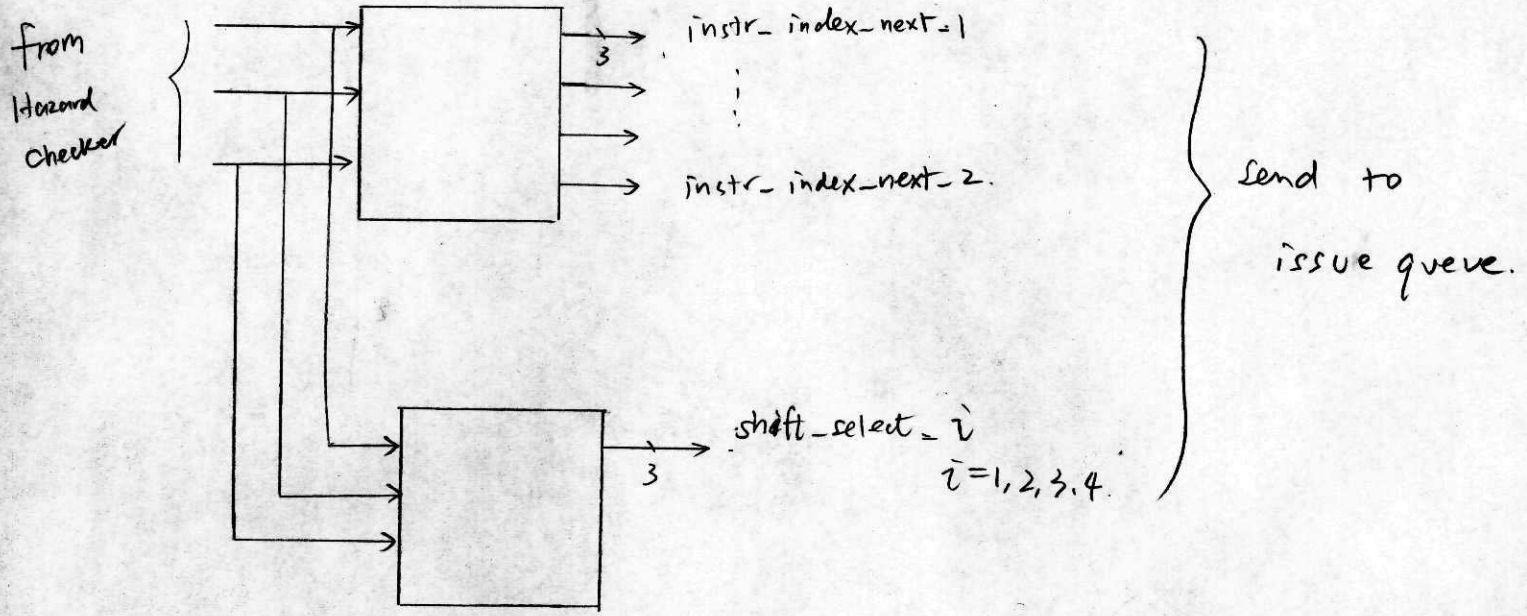
in





Stage 1 (3)

c. control logic



Stage 2 DECODE (1)

1. Basic Functionality

   ① Decode

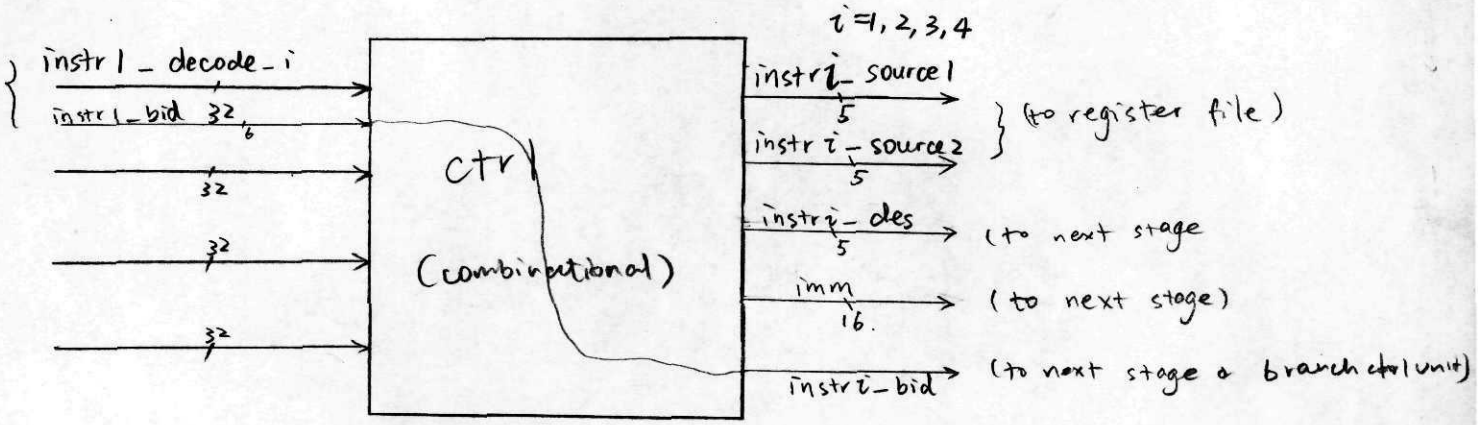
   ② Access Register File

   ③ Branch control

2. Basic Blocks.

   ① Ctrl. (combinational)

   a. input and output interfaces



   b. basic logic

   if (ADD) $i=1,2,3,4$

   instr i_decode_i (16~20) → instr i_source1

   instr i_decode_i (0~5) → instr i_source2.

   instr i_decode_i (21~25) → instr i_des

   if (BNE)

   instr i_decode_i (16~20) → instr i_source1

   instr i_decode_ (21~25) → instr i_source2

   x x x → instr i_des

Stage 2 DECODE (2)

if (LD)

instri-decode-i(16~20) → instri-source1.

× × × → instri-source2.

instri-decode-i(21~25) → instri-des.

if (SD)

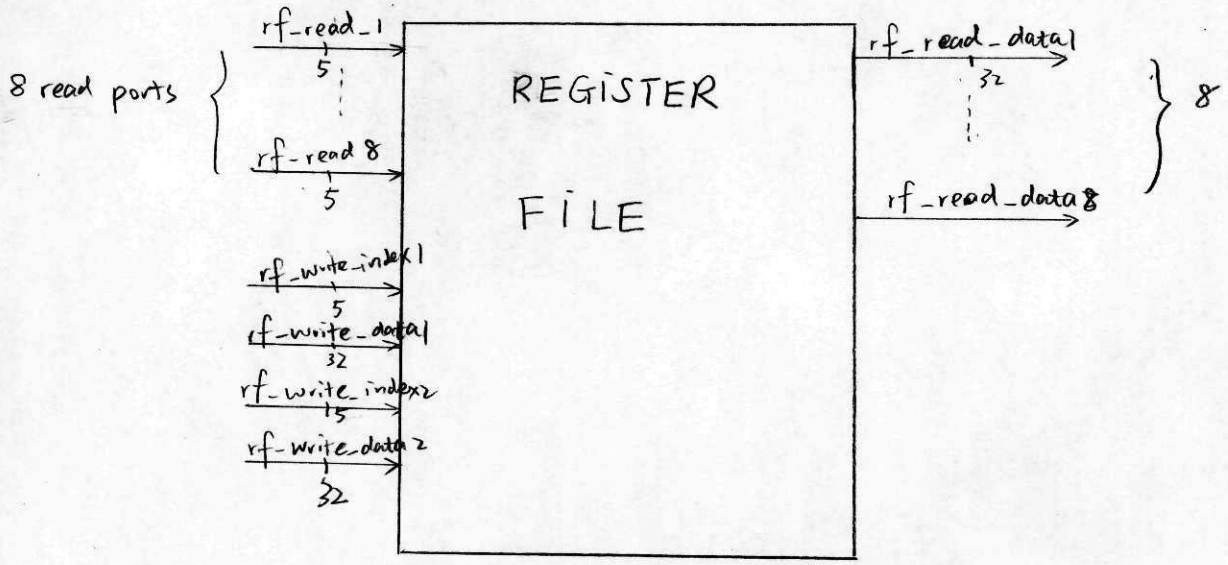
instri-decode-i(16~20) → instri-source1

instri-decode-i(21~25) → instri-source2

×××× → instri-des.

② Register File.

a. input & output interfaces



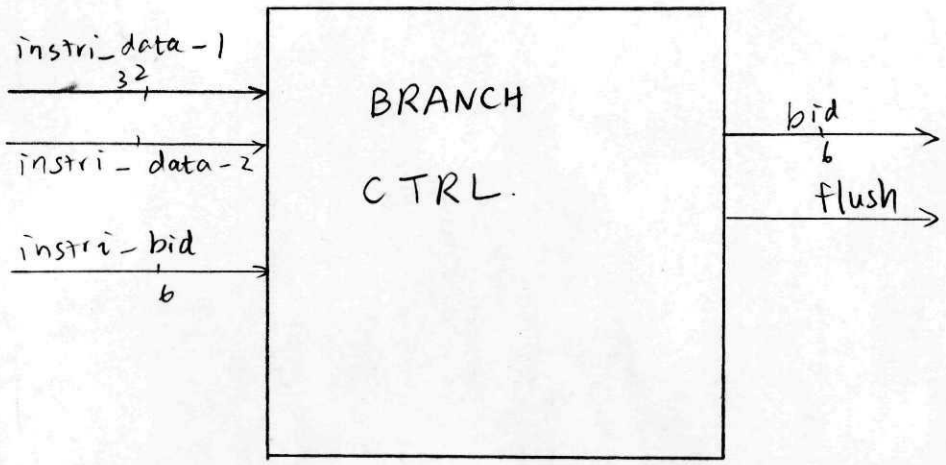
16 entries.

b. micro architecture.

Stage 2 DECODE (4)

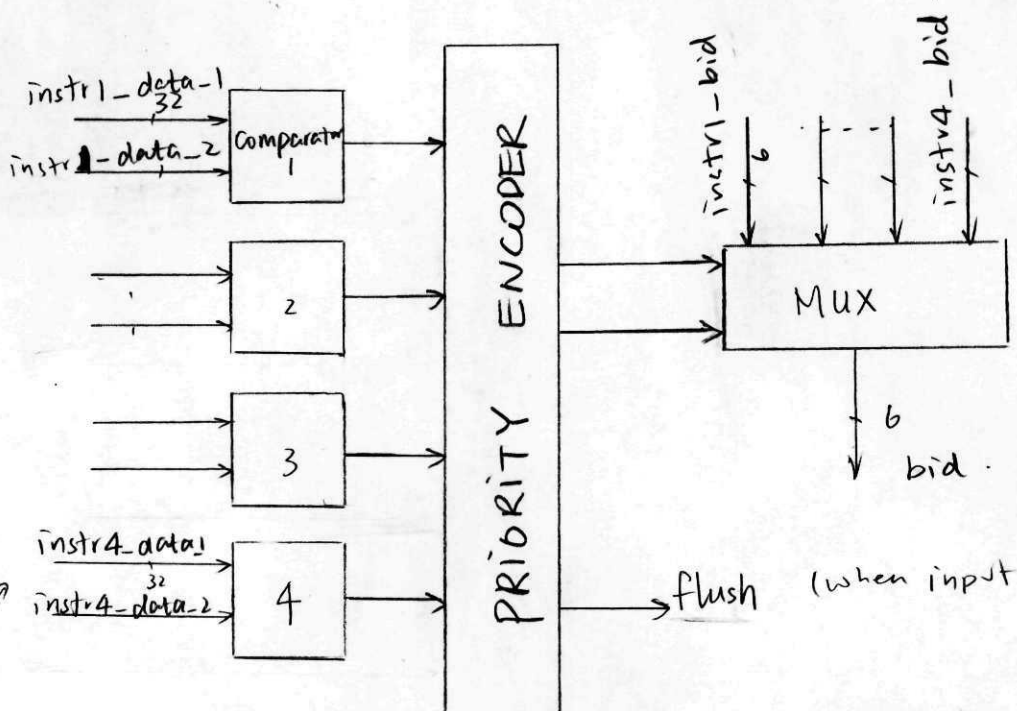
③ branch control unit.

a. input and output interfaces.



bid 6 bits.
What if # branches > 64 ?

b. microarchitecture



(when input 0000, flush = 0

① if instri is not a branch, set these data bits 0. branch not taken, predict correctly.

② if equal, output 0.

③ if more than 1 branches mispredict, use priority encoder to choose the first mispredicted branch, set flush signal and flush all the instructions after that branch.

Stage 3 EXECUTION

1. Basic Functionality.

   ① For LD/SD instructions, add the imm and the source to get the memory address

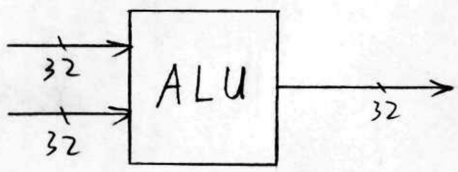
   ② For ADD instructions, add the data from two source registers to get the result for the destination register.

   ③ To speculate the branch, store the data, corresponding destination and branch id. After a branch is resolved, if it predicted correctly, the data can be committed, otherwise these data should be flushed.

2. Basic Blocks.

   ① ALU

整个 pipeline exception.



if overflow

shutdown

Stage 3 (2)

② Buffer.

a. input & output interfaces.