Multicycle implementation mMIPS

- Why multicycle ?
  - drawbacks of single cycle datapath impl. of mMIPS
    - a low latency instruction such as "jmp"
      - would've computed the result early during clock cycle
      - but long wait until end of clock cycle

- Natural improvisation
  - discretize time into smaller duration clock cycles
    - faster frequency clock signal
  - different instructions would require different number of such short clock cycles
    - eg "load word" — 5 clock cycle
      "jmp" — 2 clock cycles

Edge-triggered synchronous sequential ckt to implement MIPS

- talking in terms of clock cycles
- updates of 'state' of computation
  only happening at the end of the clock cycle (at the triggering edge of clock)
- 'state of computation' is stored in registers, memory blocks..

FSM + datapath

- multicycle MIPS
  - case study of FSM + Datapath

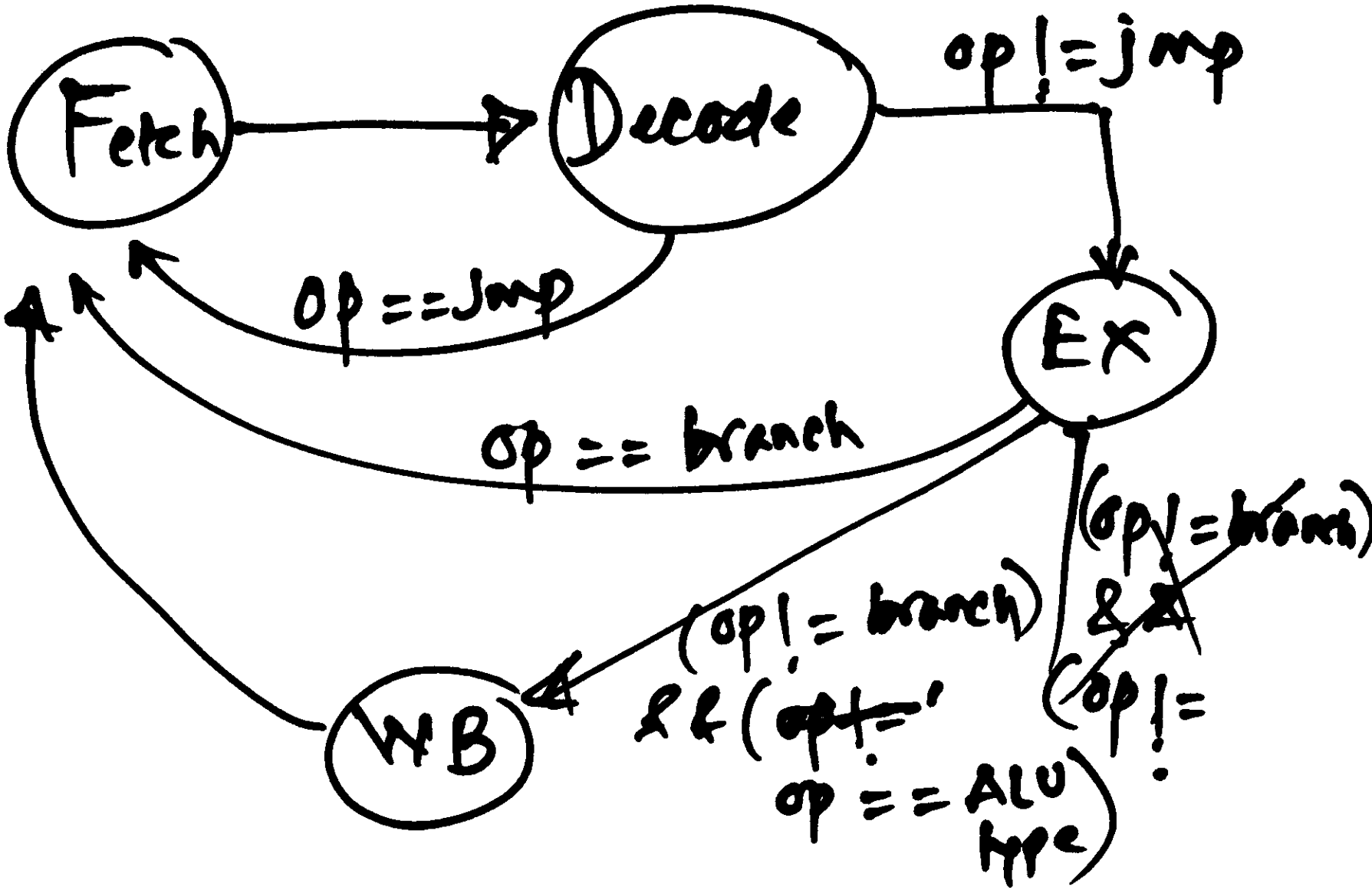
Instruction cycle : sequ. of clock cycles during which an instruction is getting processed

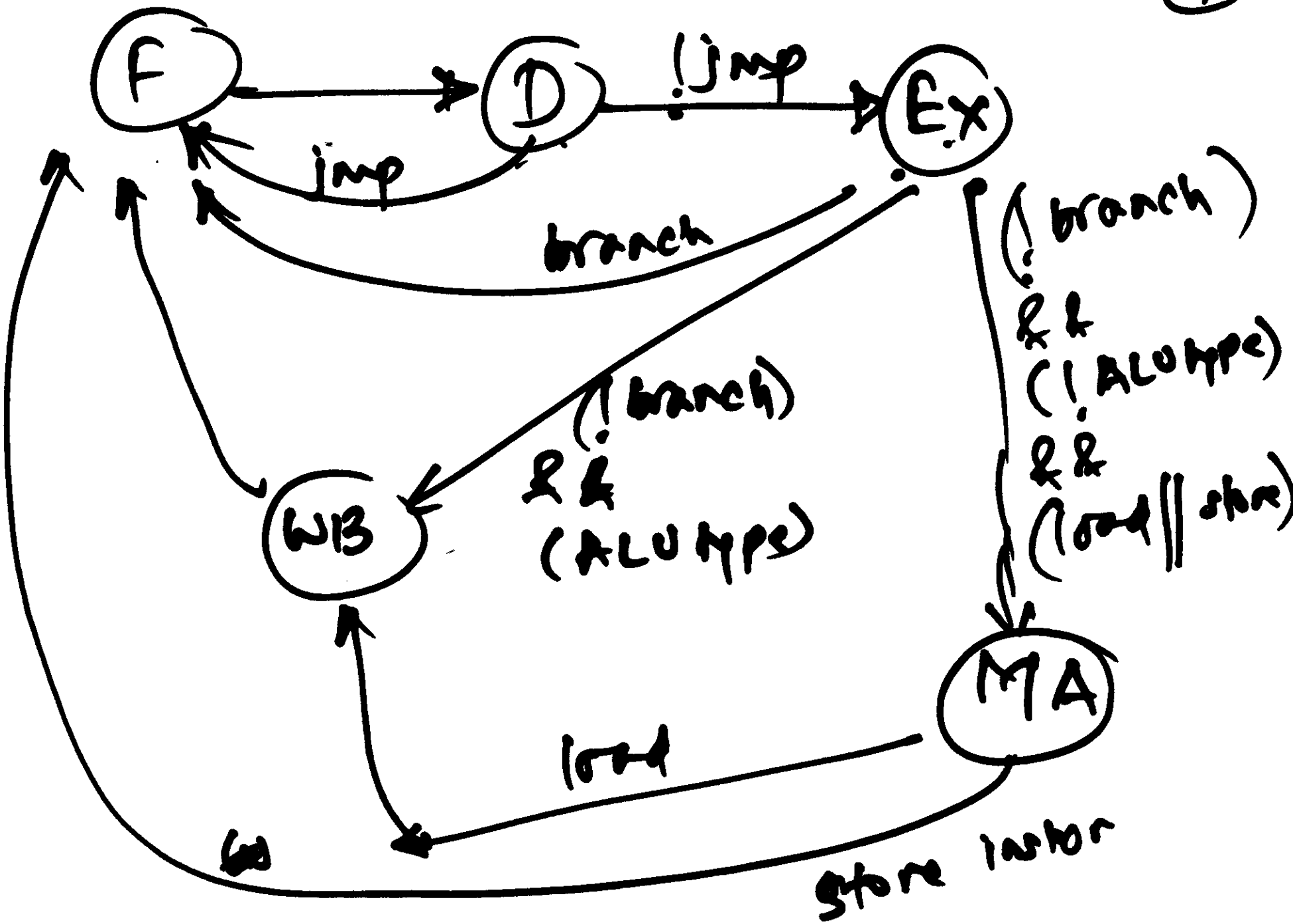
Instruction subcycle : individual clock cycles of instr-cycle.

Instruction-subcycle

- suboperations
  - fetch
  - Decoding & reading operands from Reg File
  - Arith & logic execution
  - Memory access
  - Write Back

FSM diagram

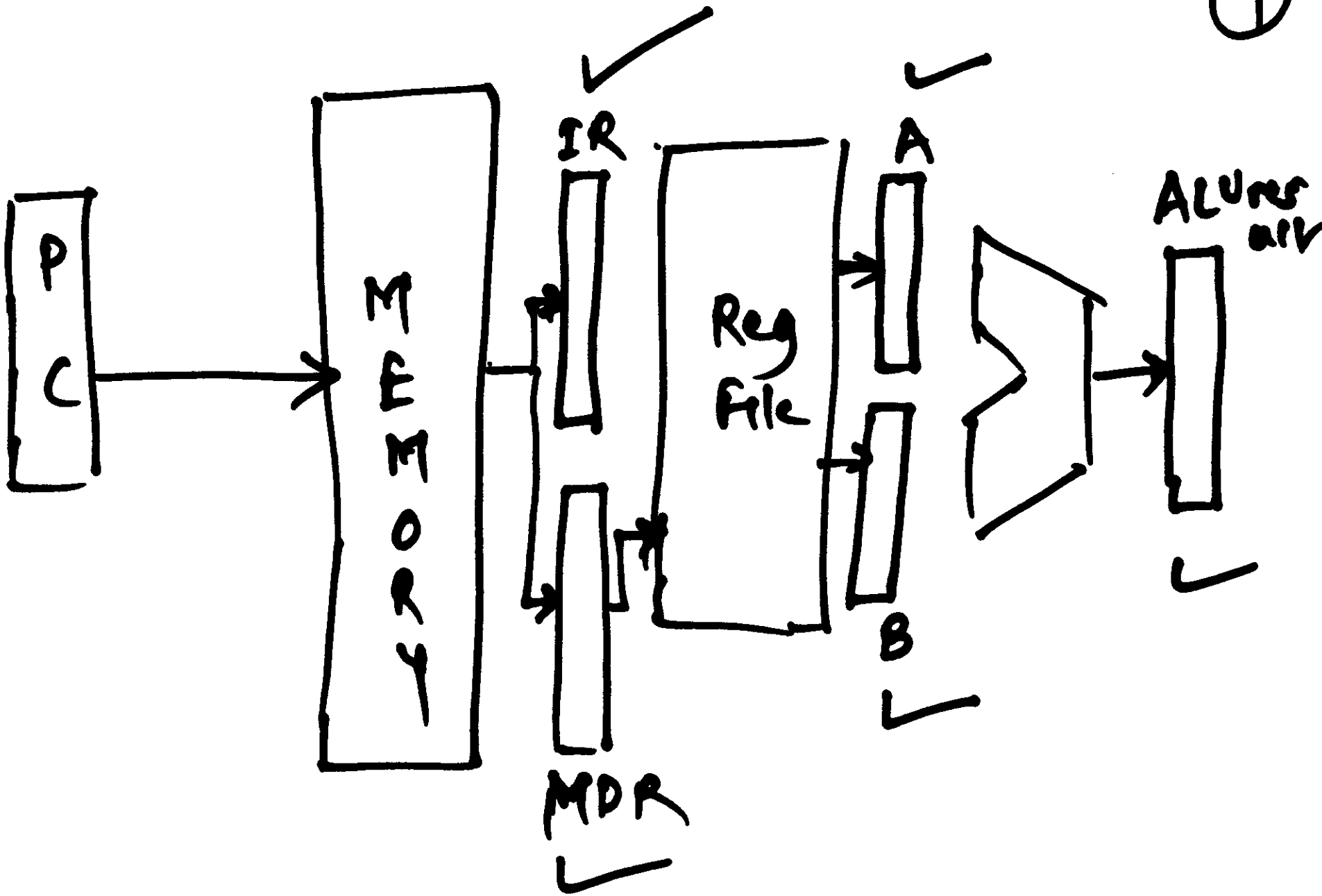

Fetch
Decode
op != jmp
op == Jmp
Ex
op == branch
(op != branch)
&&(op !=
op == ALU type)
(op != branch
&&
(op !=
WB


F
D
!jmp
Ex
jmp
branch
(!branch)
&&
(ALUtype)
(!branch)
&&
(!ALUtype)
&&
(load || store)
WB
MA
load
store instr

Datapath for multicycle implem.

Which datapath components of mMIPS ?

- PC
- ALU
- Memory (We'll see that single memory for I + Data)
- Reg file
- ? what more ? – for storing results of subopterations

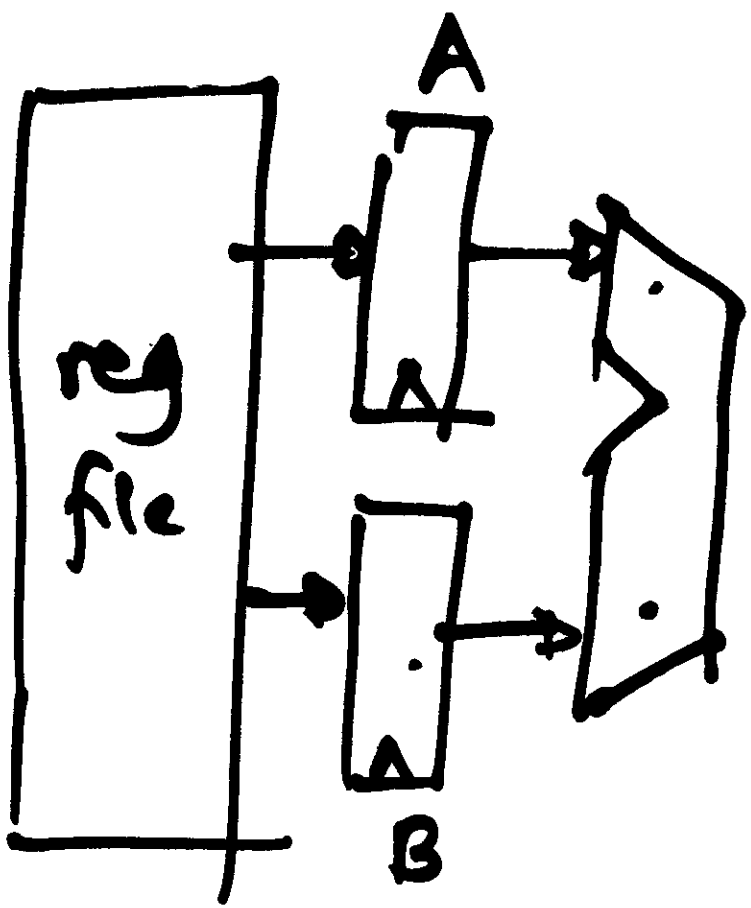

PC
MEMORY
IR
MDR
Reg File
A
B
ALUMR
AIV

More registers for multicycle MIPS:

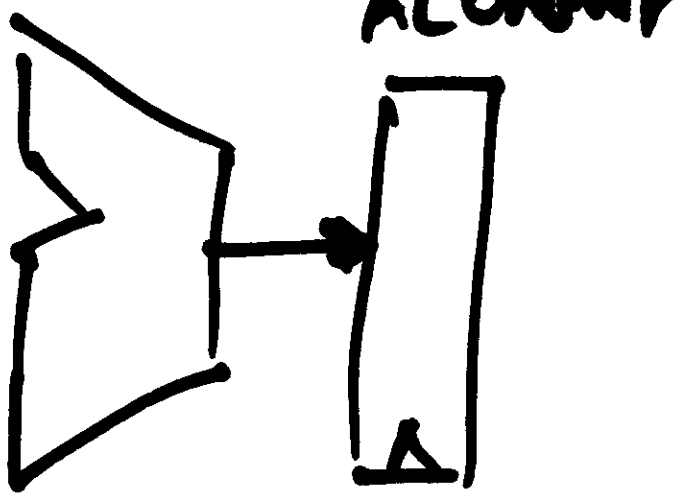
- IR : for storing/registering the result of ~~F-stak~~ Fetch subcycle
- MDR (memory data register)
  : storing result of suboperation MemAccess during instruction cycle of "load word" instruction

A register

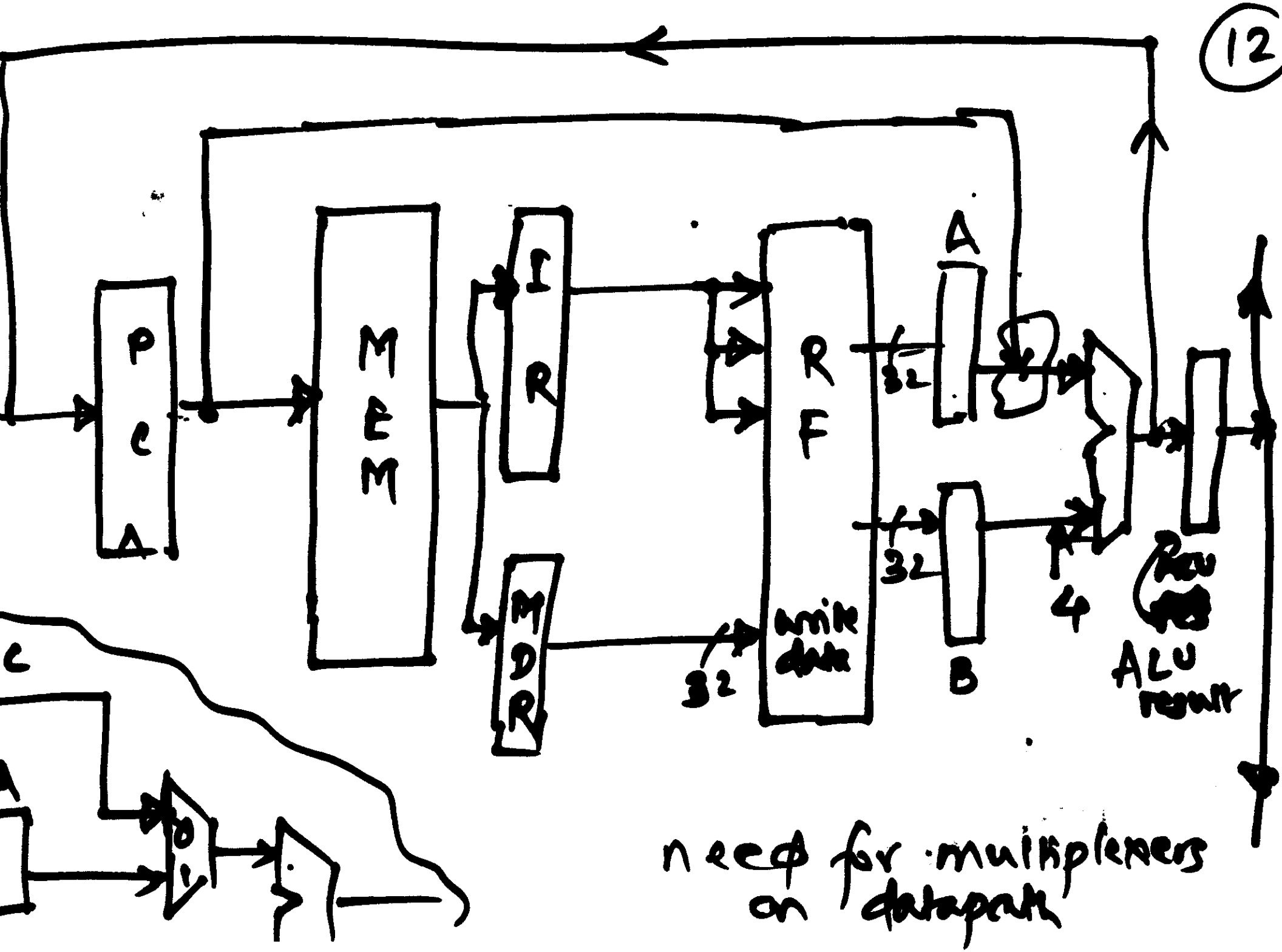
B - register

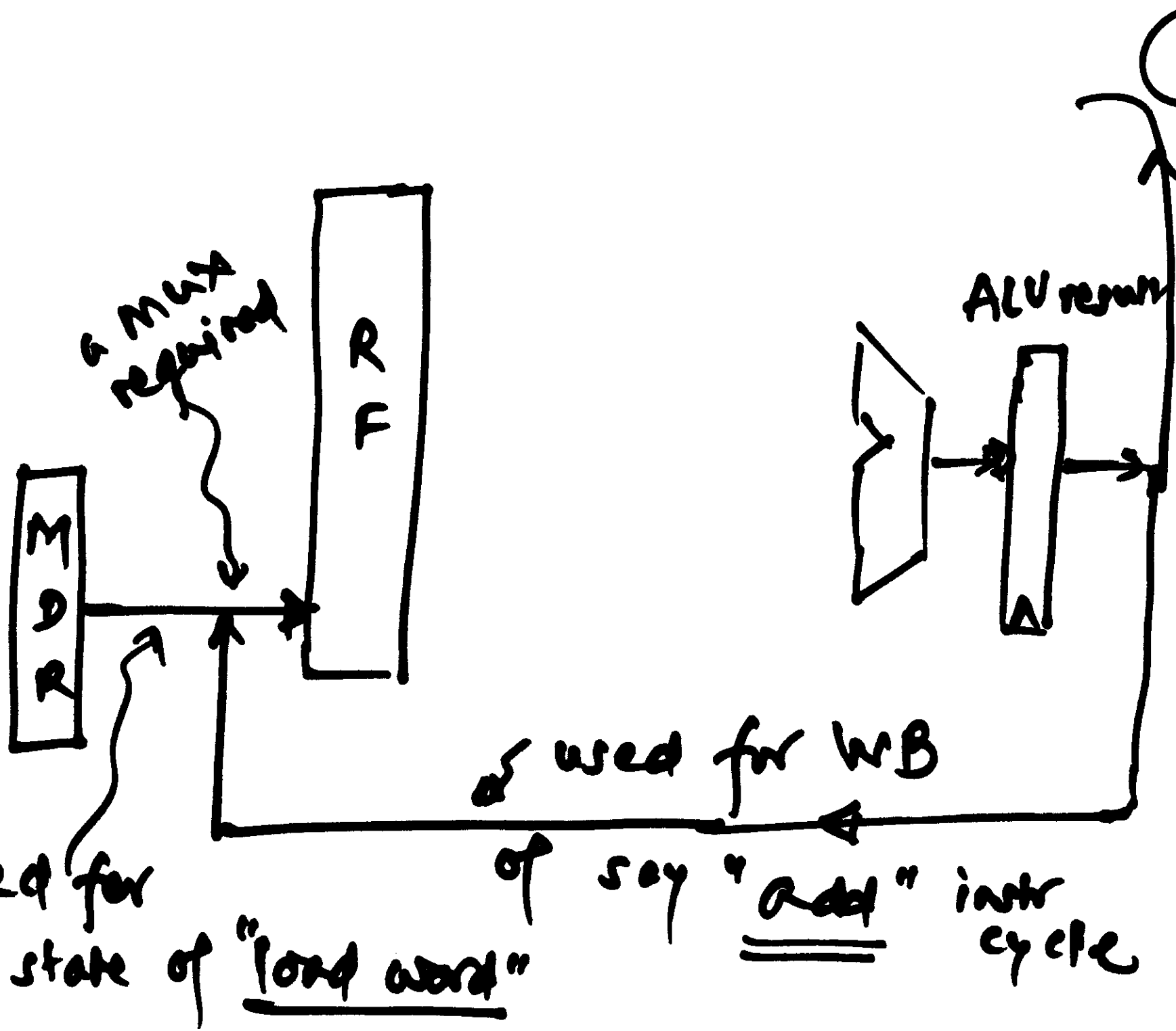


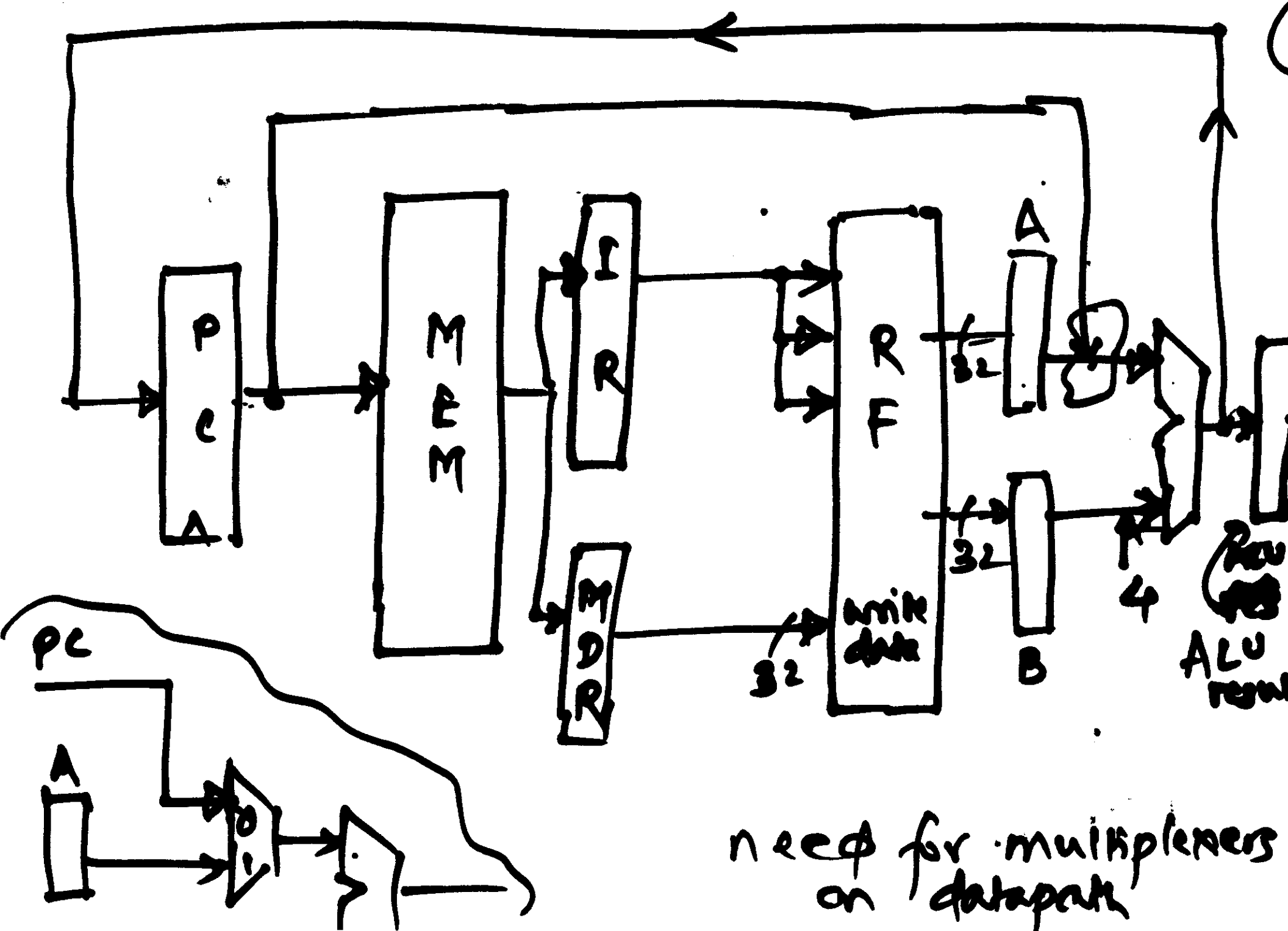
ALU result



: to store result of
Ex suboperation


P
C
A
M
E
M
I
R
M
D
R
R
E
G
F
write data
32
32
32
A
B
4
ALU
result
C
0
1
need for multiplexers
on datapath


a MUX required
RF
MDR
ALU result
of used for WB
of say "add" instr cycle
used for WB state of "load word"



need for multiplexers on datapath