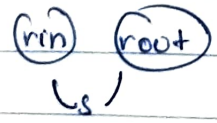
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_numeric_std.all;


package cpu_4004_pkg is
    subtype byte is integer range 0 to 15;
    constant byte_values : integer := byte'high + 1;
    subtype addr_range is integer 0 to 15;
    type memory_type is array (addr_range) of byte;
```

registers [process: register] (clk, rst).
flags
ip
os..



classmate
Date 19.4.2020
Page 1

① make a simple 16-bit CPU (using behavioural design).

① load rd, [rs+imm] } mem.
① store rs, [rd+imm]

mov rd, rs
?

and rd, rs
or rd, rs
not rd
xor rd, rs
} logical

add rd, rs
adc rd, rs
sub rd, rs
sbb rd, rs

mul rd, rs
imul rd, rs
div rd, rs
idiv rd, rs.
} arithmetic

control

je
jne
jl
jle
jg
jge

jz rds
jnz rs
jgz rs
jlz rs
① jmp rs+imm
call ?
ret ?
?

no stall.

① fetch only. (instruction / bytes). (16 bits)

② fetch + decode only (instructions). (simulate)

③ fetch + decode + execute (some)

④ fetch + decode + execute all

↓

breakup to components.

↓

64 bit.

two process design