

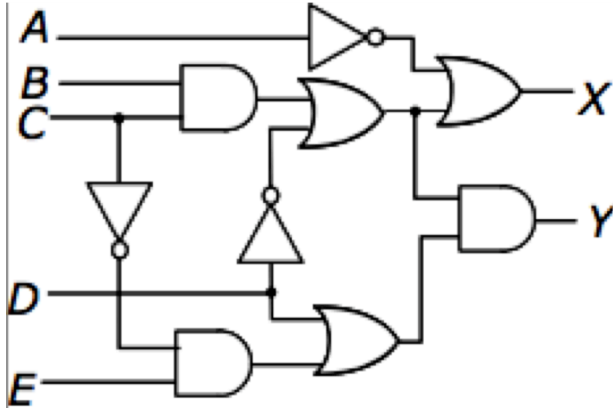
Homework 1 (Due Apr. 8 before class)

Problem 1:

Write an entity declaration and a behavioral architecture body for a two-input multiplexer, with input ports 'a', 'b' and 'sel', and an output port 'z'. If the sel input is '0', the value of 'a' should be copied to 'z', otherwise the value of 'b' should be copied to 'z'.

Problem 2:

Consider the circuit shown below. Write the behavior models for the NOT gate, the AND gate, and the OR gate. Using the VHDL models of the three gates, write a structural model for the whole circuit.



Problem 3:

Consider the following architecture description.

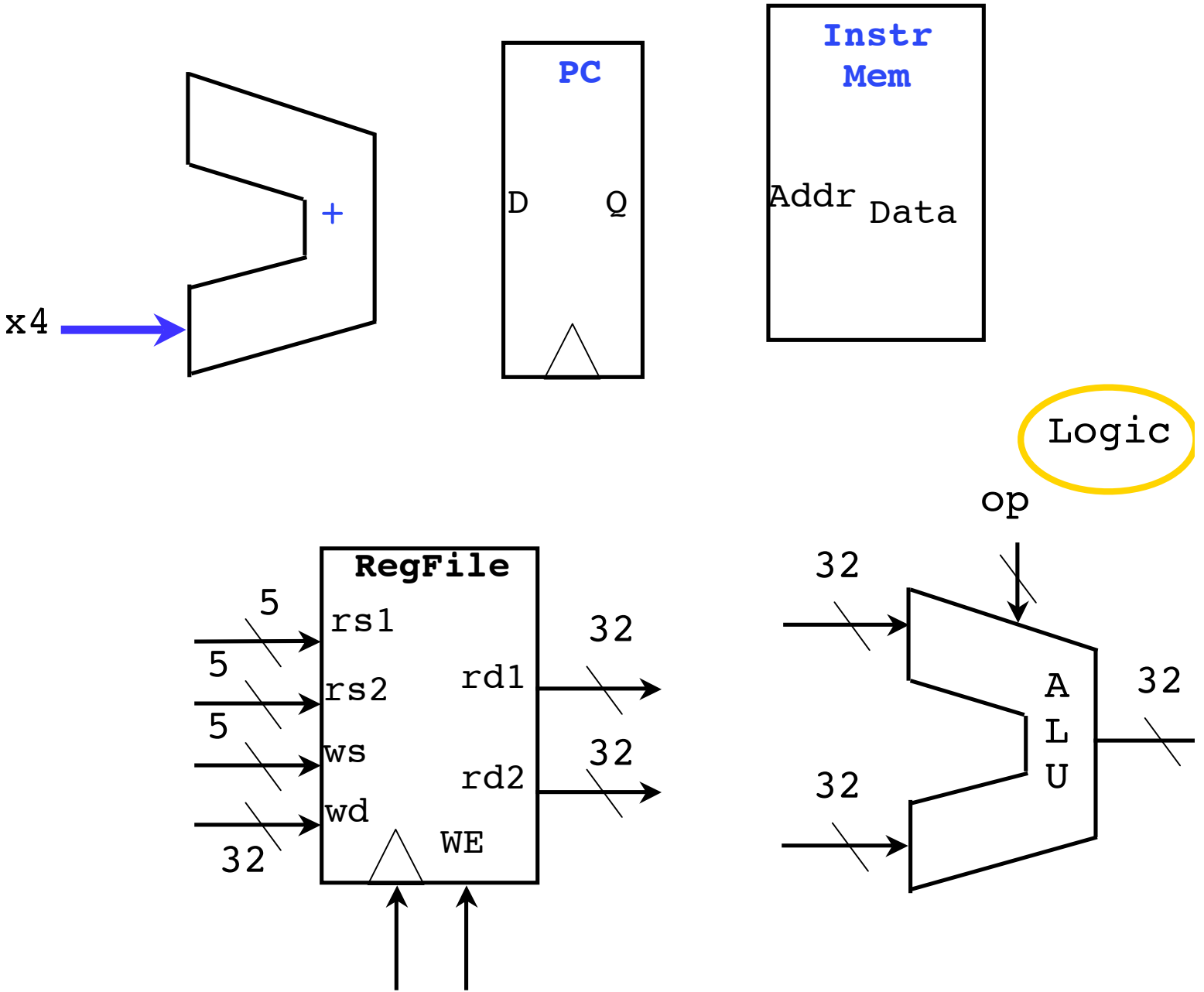
```
architecture XYZ of ABC is
signal a, b, c, d: integer := 1;
signal clk: bit := '1';
begin
clock: process is
begin
clk = '1','0' after 4 ns;
wait for 8 ns ;
end process clock;
foo: process is
variable x, y, z: integer := 1;
begin
-- point 1
x := y + 2*(b**3);
a = b**2 + x;
b = a**2 + y after 4 ns;
--point 2
wait for 2 ns;
--point 3
c = a+b;
y := 2*a + c;
--point 4
wait until clk'event and clk='1';
d = a + b + c;
--point 5
wait on clk;
end process foo;
end architecture XYZ;
```

Determine the simulation time, and values of the signals and variables at the following points and iterations given in the table below. Assume that simulation begins at time 0 and that the first value of the clock is '1' (and thus that the first transition on the clock will be negative edge and will occur at time 4 ns—half the clock period).

Iter 1, point 1	Time	a	b	c	d	x	y	z
Iter 1, point 2	Time	a	b	x				
Iter 1, point 3	Time	a	b	x				
Iter 1, point 4	Time	b	c	y				
Iter 1, point 5	Time	b	c	d				
Iter 2, point 1	Time	a	b	c	d	x	y	z
Iter 2, point 2	Time	a	b	x				

Problem 4:

Draw the data path that simulates the execution of a R-format instruction *add* in a single-cycle CPU. Label the edges with the logic steps, ie., the order in which the operations happen. The first step starts from the program counter PC, and should be labeled "1".



Problem 5:

Add a second write port *WS2* and *WD2* to the register file shown below. Assume that *ws2* will always have a different value with *ws*. Draw your modification to the circuit. You may modify only one register, if the identical changes will be repeated to other registers. Each register still has only one 'E' port and one 'D' port, which means that you must share the 'E' (enable) port and the 'D' (write data) port between two write ports. You must avoid any data conflict on the shared 'E' and 'D' ports.

