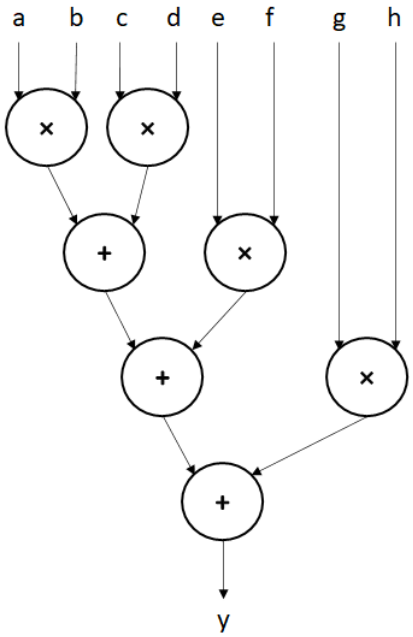
Projecto de Sistemas Digitais
Digital Systems Design
MEEC, 2021/22, 2nd Exam

Table 1: Timing characteristics and resource utilization of logic-arithmetic components				
	Area	t_P	t_{SETUP}	t_{HOLD}
16-bit Register	10	2 ns	1 ns	1 ns
BRAM	100	2 ns	1 ns	1 ns
16-bit Adder / Subtractor	16	5 ns		
16-bit Multiplier (simple)	120	14 ns		
16-bit Multiplier (2-stage pipeline: 1 internal register-level)	130	8 ns		
16-bit Divider	500	85 ns		
MUX 2:1 (16-bit data words)	16	3 ns		
MUX 4:1 (16-bit data words)	32	4 ns		
MUX 8:1 (16-bit data words)	64	5 ns		
16-bit logic operators	16	3 ns		

Note: all answers must be properly justified, without which they will not be classified.

1. **[3 val]** Consider the data flow graph in the figure on the right.



Define a priority list using the critical path as metric.

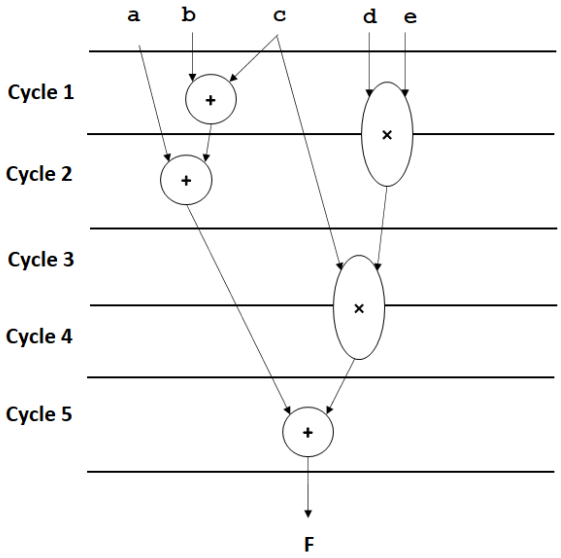
Consider that

- the inputs are stored in independent registers
- at the end of the calculation the output y is stored in a register
- the arithmetic operators available are 2 multipliers (simple) and 2 adders
- the clock period is 25 ns.

All input values are 16-bit integers in 2's complement representation and all operators, buses and datapath registers are 16-bit wide with the timing/area characteristics shown in table 1.

Obtain a list scheduling using the priority list defined.

2. Consider the schedule defined in the figure, at the right, for a circuit whose datapath is to be implemented with one adder and one multiplier (simple).
   All input values are 16-bit integers in 2's complement representation.
   All operators, buses and datapath registers are 16-bit wide, with the timing/area characteristics shown on the 1st page.
   All circuit registers must be positive edge-triggered, fully synchronous and be synchronized with a unique global clock.



a) **[4 val]** Design the circuit datapath to use the smallest possible **area** (given the constraints indicated).
   Indicate the number of registers and multiplexers required.
   Draw the block diagram of your datapath.
   Estimate the area occupied, the minimum clock period, and the latency for your design.
   Justify all the bindings and sharing of resources, and **explain why the area estimated is the smallest possible**.

b) **[1.5 val]** Draw the state diagram of the control unit of the circuit for the datapath designed in (a).
   Indicate explicitly the values of the control output signals for each state. Justify.

3. **[3 val]** Consider the (square of the) Euclidean distance, *D(p,q)*, for calculating the distance between two points *p* and *q* in 4 dimensions:

$$D(p,q) = (p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + (p_4 - q_4)^2$$

The p_i and q_i input values are 16-bit integers in 2's complement representation.
Consider also that you want to perform a sequence of distance calculations
from a given ***p*** point (stored in a 64-bit register) to a set of 100 ***q*** values,
which are stored in one independent 100×64-bit memory with 2-ports for reading.
Sketch the draft block diagram of the datapath of a pipelined architecture, that executes the whole calculations **as fast as possible**.
Consider for all components the timing/area characteristics shown on the 1st page. All circuit registers must be positive edge-triggered, fully synchronous and be synchronized with a unique global clock.
What is the minimum clock period that you can use, what is the maximum throughput that you can achieve and what is the minimum number of multipliers, adder/subtracters and intermediate pipeline registers required to achieve it? Justify.

4. Design an arithmetic component to perform the operation Y = 4.5 × (A - D) + C,
   where the operands are real numbers represented in fixed-point format, as indicated:

Signal	*Fixed-point format*
A	Q12.4
D	Q2.14
C	Q30.2

a) **[2 val]** Define the fixed-point representation format required for the internal and output signals:

```
s = a – d ;    -- output of the subtracter
m = s * 4.5 ; -- output of the multiplier
y = m + c ;     -- output of the adder
```

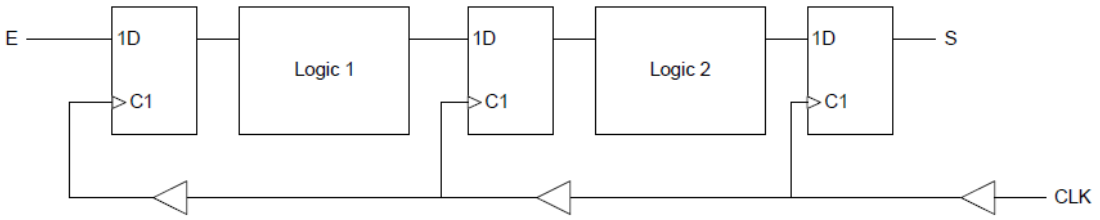
to avoid the existence of overflows and to avoid any loss of precision in the computations. Justify.

b) **[2 val]** Complete the pseudo-VHDL specification (in the last page) of the architecture of the component to perform the required operations (don't care about minor syntax issues).
Define and dimension all signals, operators, and buses to preserve maximum precision in the calculations and avoid overflows. The output ports are ***yi*** (integer part of ***y***) and ***yf*** (fractional part of ***y***).

c) **[1 val]** Consider now that the arithmetic operations of the component will be implemented using DSP48E1 blocks. How many DSP48E1 blocks will be required? Justify.

d) **[1 val]** Modify the specification defined in b) to include input and output registers, and the pipeline registers necessary for the circuit to execute as fast as the technology allows.

5. Consider the following circuit.



The circuit components have the following timing characteristics:

t_{PFFmax}= t_{PFFmin}=4ns; t_{SETUP}=2ns; t_{HOLD}=1ns;
$t_{PLOG1max}$=9ns; $t_{PLOG1min}$=3ns; $t_{PLOG2max}$=10ns; $t_{PLOG2min}$=5ns;

a) **[1 val]** Estimate the minimum clock period for the circuit to work correctly, considering the *buffers* in the clock network "ideal" ($t_{Pbuffers}$=0ns). Justify.

Consider now a timing analysis with the following clock constraint:

```
create_clock -add -name sys_clk_pin -period 25.0 -waveform {0 5} [get_ports clk]
```

b) **[1 val]** Estimate the WNS (Worst Negative Slack) of the circuit, considering that there exists a negative clock skew ($t_{Pbuffers}$=3ns). Justify.

c) **[0.5 val]** Estimate the WHS (Worst Hold Slack), considering that there exists a negative clock skew ($t_{Pbuffers}$=3ns). Justify.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity madda is
  Port ( a  : in  std_logic_vector (15 downto 0);
         d  : in  std_logic_vector (15 downto 0);
         c  : in  std_logic_vector (31 downto 0);
         yi : out std_logic_vector ( ??? );
         yf : out std_logic_vector ( ??? ));
end madda;

architecture behavioral of madda is
  signal as, ds : signed(15 downto 0);
  signal cs : signed(31 downto 0);

-- internal signals
  signal s : signed( ??? );
  signal m : signed( ??? );
  signal y : signed( ??? );
  signal fourdotfive : signed( ??? );
  ???

begin
 as <= signed(a);
 ds <= signed(d);
 cs <= signed(c);

 -- arithmetic operations
 ???

end behavioral;
```