

U. PORTO**FEUP** FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTOCurso MIEECData 01 / 02 / 2021Disciplina PSDIAno 4Semestre 1Nome João Alexandre da Silva Dado

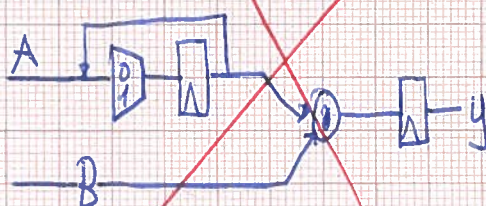
Espaço reservado para o avaliador

1: a) Na síntese RTL irá ser formado um circuito lógico, que resulta da tradução comportamental e/ou estrutural do código Verilog. Este último, mediante a forma como foi escrito, vai ter um determinado número de ciclos de relógio que não vai ser alterado com processos de otimização ocorridos nesta síntese. - Verdadeiro (V)

b) Apenas após ser realizada a ~~realizada~~ transformação "Elab. Ponte" é que teremos conhecimento dos tempos de atraso de propagação dos vários circuitos combinacionais formados entre flip-flops, e, com estas informações, pode-se obter um maior detalhe na estimativa da frequência (ou seja, apenas se obtém uma frequência máxima admitida, isto é, uma estimativa desta, após a simulação física (ps-"Elab. Ponte"). - Falso (F)

c) Após a síntese física, as ferramentas de projeto estimam valores para a frequência máxima com o pressuposto de que todos os registos são gerados pelo mesmo ciclo de relógio. No entanto, nem sempre isto ocorre: o circuito combinacional com maior tempo de atraso de propagação pode ocorrer (fazer enable) apenas após N ciclos de relógio (fazendo com $T_{delay} = N \cdot T_{clk}$), o que faz com que o circuito consiga, de facto, trabalhar a uma frequência maior do que a repetida pelas ferramentas. - Verdadeiro (V)

2: O facto de sabermos que A pode apresentar várias sequências nulas ou unitárias faz com que possamos "desligar" A durante estas longas sequências nulas, poupando na energia consumida:



φ

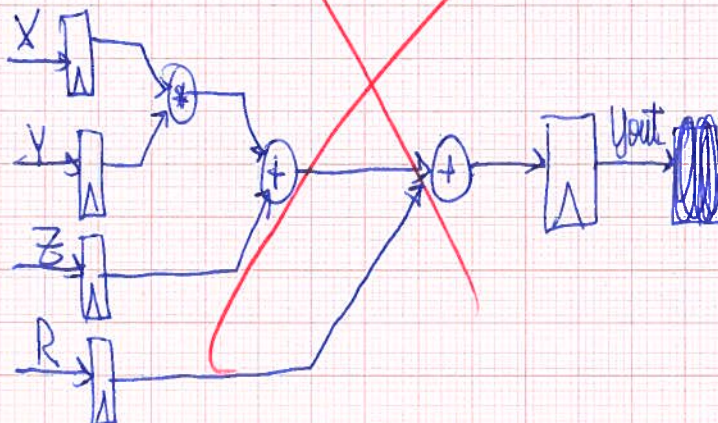
```
3: a) modulo soma complexa {input clock;
    input en; // enable
    input [17:0] X, Y, Z;
    input R;
    output [17:0] Yout
};
```

```
int i;
reg [17:0] rX, rY, rZ;
reg rR;
reg [17:0] rXY, rXYZ, rsuma;
assign @*
begin
    if (reset)
        // todos os registros são inicializados a zero
    end
    else
        begin if (en)
            rXY = rX * rY;
            rXYZ = rXY + rZ;
            rsuma = (rXYZ 4096 ? rXYZ : rXYZ + 64'000);
        end
end
assign Yout = rsuma;
endmodule
```

```
assign @(posedge clock)
begin
    for (i = 0; i < N; i++)
        begin
            R = R[i]
        end
end
```

φ

b) Admitindo que o controlador que define o valor de R será ajustado, para eliminar a necessidade de realizar o ciclo de espera entre inputs, o circuito poderia ser modificado eliminando o mux seguinte ao segundo somador e o registro final



Ⓞ c) Dado que este circuito usa dois sinais de relógio com frequências diferentes, torna-se importante garantir a sincronização destes sinais. ~~X~~

~~Se introduzirmos um único relógio~~

Uma forma de garantir isso mesmo seria um clock geral formado por ambos, mas mesmo esta opção pode criar problemas.



?!"

U. PORTO**FEUP** FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTOCurso MIEECData 01 / 02 / 2021Disciplina PSDIAno 4 Semestre 1Nome mate Alessandra da Silva e Silva

Espaço reservado para o avaliador

5: a) i) A operação ~~produto aritmético~~ vai calcular o resultado do produto, calculando um bit por iteração, pelo que o multiplicador apenas necessitaria do número de ciclos de relógio referente ao número de bits (neste caso, $64 + 1/2$ ciclos/ciclo de relógio para arredondamentos, etc.), o que ~~permitiria~~ um decréscimo de complexidade lógica do circuito.

ii) Calculando um bit por iteração fará com que os tempos de atraso de propagação entre as 64 primeiras combinações formadas entre flip-flops sejam menores que o clock em várias iterações, aumentando, assim, a frequência máxima de relógio.

b) a-i) Uma menor área ~~para esse módulo~~ ~~por um menor~~ iria aumentar a frequência de relógio. +!

a-ii) Uma maior frequência de relógio, suportada por este módulo isoladamente, não teria efeito na área total do circuito.

