AT 1 - Amanda Quirino R. Dos Santos

01) $F = A.B + A.B.C'.D + A.B.D.E' + A.B.C'.E + C'.E.D$

$F = A.B(1 + C'.D + D.E' + C'.E) + C'.E.D$

→ Como $1+X=1$ sempre, então:

$$\boxed{F = A.B + C'.E.D}$$

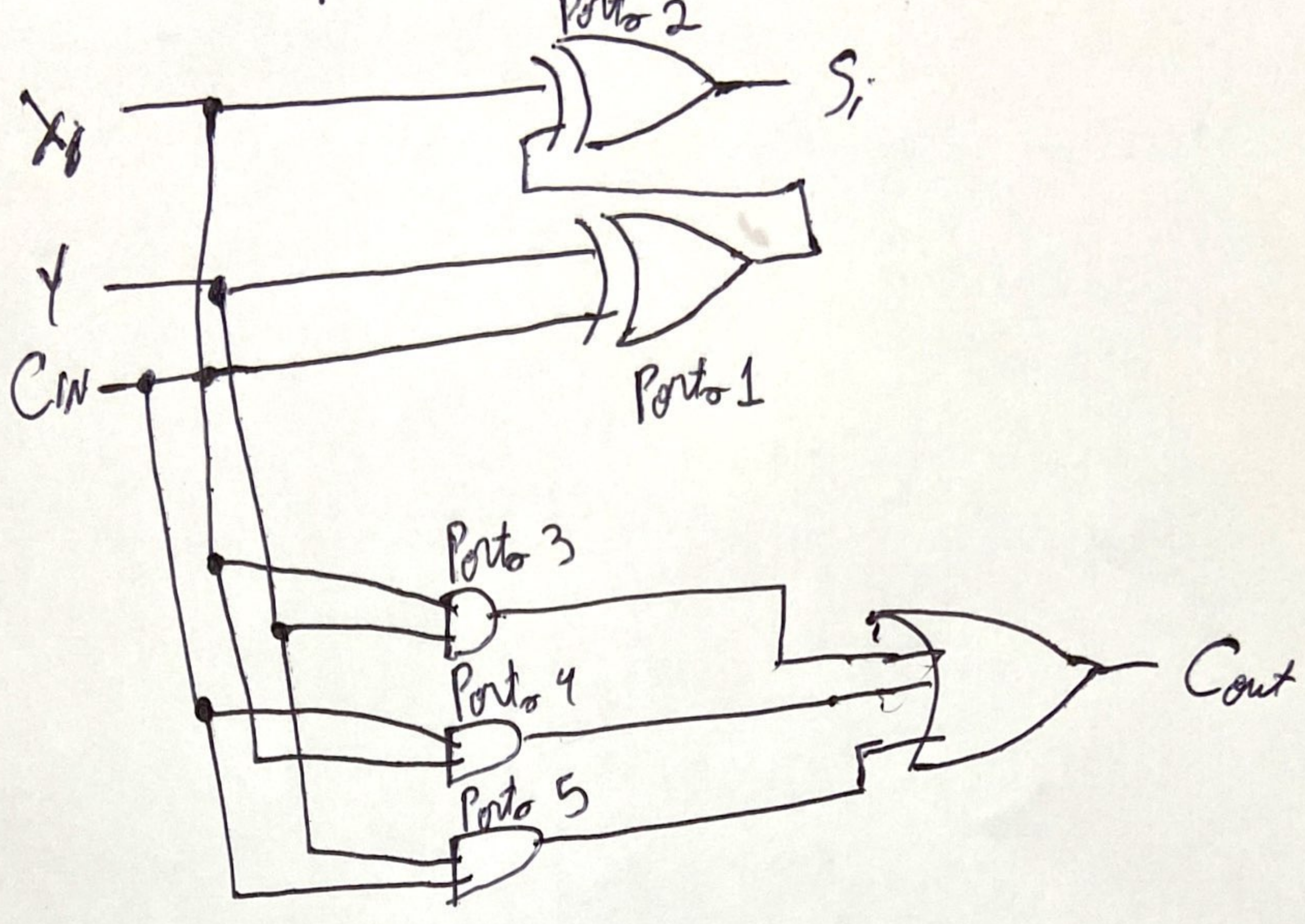
02) Falso, se os implicantes maiores incluírem mintermos para os quais $h=1$, pois caso $X=0$, os mintermos correspondentes a eles não serão incluídos na simplificação, gerando a função mais simplificada possível.

05) Falso, pois um multiplexador 4-para-1 tem 2 pinos para sinais de Controle e $2^2 = 4$ entradas.

Também chamado de linhas de Controle ou bits de seleção

06) Um decodificador de BCD para 7 segmentos possui 3 pinos de entrada (que serão ligados a B, C e D) e 8 pinos de saída, sem considerar a entrada do ENABLE.

07) Somador Completo Circuito:



Logo CIN = 0, X = A, Y = B e CIN = 0 e [redacted] F = Cout,

pois como Porta 4 = Porta 5 = 0, pois CIN = 0, então Cout = Porta 3 = A.B

e S_i = ~~[illegible]~~ A Xor B, pois como CIN = 0, então Porta 1 = B, então

S_i = Porta 2 = A Xor B

09) 127 |2
1 63 |2
1 31 |2
1 15 |2
1 7 |2
1 3 |2
1 1

Logo $(127)_{10} = (\overbrace{1111111}^{7\ bits})_2$, logo $(-127)_{10} = (0000001)_2$, de acordo com a regra prática.

Como queremos fazer com 8 bits, então:

$(127)_{10} = (\underset{\downarrow}{0}1111111)_2$ e $(-127)_{10} = (\underset{\downarrow}{1}0000001)$

Pois 127>0 — Pois -127<0

Já: 255 |2
1 127 |2
1 63 |2
1 31 |2
1 15 |2
1 7 |2
1 3 |2
1 1

Portanto $(255)_2 = (1111111)_2$, que já tem 8 bits

Já $(128)_2 = (1000000)_2$, pois $2^7 = 128$, portanto

$(-128)_2 = (1000000)_2$ de acordo com a regra prática

logo para ter 8 bits temos $(128)_{10} = (01000000)_2$ e $(-128)_2$

$= (10000000)_2$, porém $(10000000)_2 = -0$ em complemento de 2, logo não é possível ser representado.

10) Um identificador em verilog deve começar com letra ou o _ , então nota-se que os identificadores 123A_f e $A123_ não são identificadores legais em Verilog.

Além disso palavras reservadas em Verilog também não podem ser identificadores legais, logo o identificador and também não é

Dessa forma, resta apenas os identificadores A_123, C_1_C_2 e and1 como identificadores legais em verilog.

8) $X < (Y + B_{in})$, então $B_{out} = 1$

logo se $X = 0$ e (~~Y = 1 e Bin = 1~~ $Y + B_{in} = 10$ ou 11) ou se $X = 1$ e $Y + B_{in} = 11$

$Diff = X - Y - B$.

logo ~~se a X = 0 e~~ Temos a seguinte tabela verdade

X	Y	Bin	Bout	Diff	
0	0	0	0	0	$\rightarrow 0-1 = 1$ e $1-1=0$
0	0	1	1	1	$\rightarrow 0-0 = 0$ e $0-1 = 1$
0	1	0	1	1	$\rightarrow 0-1 = 1$ e $1-0 = 1$
0	1	1	1	0	$\rightarrow 0-1 = 1$ e $1-1 = 0$
1	0	0	0	1	$\rightarrow 1-0 = 1$ e $1-0 = 0$
1	0	1	0	0	$\rightarrow 1-0 = 1$ e $1-1 = 0$
1	1	0	0	0	$\rightarrow 1-1 = 0$ e $0-0 = 0$
1	1	1	1	1	$\rightarrow 1-1 = 0$ e $0-1 = 1$

$0 < 1$ (linhas 0 0 1 e 0 1 0)

$0 < 2$ (linha 0 1 1)

~~Regra prática = 00 = 11~~

~~Diff → X Y Bin = 000 = 0 1 (11)~~ =

→ 0 não é menor que 00

→ $0+1 = 01$ e $0 < 01$

→ $0 < 01$

→ $1+1 = 10$ e $0 < 10$

→ $0+0 = 00$ e $1 > 00$

→ $0+1 = 01$ e $1 = 01$, logo Bout = 0

→ ~~10 ZOP~~ $1+0 = 01$ e $1 = 01$, logo ↑

→ $1+1 = 10$ e $1 < 10$, logo Bout = 1