

---

---

RA:	Nome:	Entrega
Lista de Exercícios N° 5		(ver site)

Assunto coberto: capítulo 5 (circuitos aritméticos), uso do Quartus II, capítulo 6 (multiplexadores e decodificadores). Questões com uso do Quartus: Q1a, Q1b, Q2, Q3.

## Q1.

A comparação aritmética entre dois números é uma operação muito utilizada tanto em circuitos dedicados quanto em processadores. Em processadores, ela é fundamental para implementação de fluxos condicionais nos programas em alto nível, como *if*, *for*, *while*, etc. Existem inúmeras formas de se projetar um comparador, apesar da maioria das formas serem ineficientes ou deficientes em aspectos como desempenho, área ou complexidade.

### 1.1

Em circuitos dedicados, é importante ter projetos bastante regulares e uniformes, para facilitar o layout físico. Por esse motivo, normalmente busca-se a implementação utilizando-se a abordagem *bit-slice*, em que projeta-se o circuito para um bit e replica-se o circuito para todos os bits. O somador/subtrator *ripple-carry* foi projetado desta maneira.

**Projete e simule** (com todas as possíveis entradas) um comparador utilizando a metodologia *bit-slice*, que tem como entrada dois **naturais** de 4 bits  $A(3:0)$  e  $B(3:0)$  e como saída 3 sinais *gt*, *lt* e *eq*, que significam respectivamente  $A > B$ ,  $A < B$  e  $A = B$ .

Observações:

Quantos são os sinais de comunicação entre dois comparadores de **um** bit? Será um sinal suficiente, como é no somador ripple-carry?

Qual é fluxo do sinal entre os comparadores de **um** bit? Será que o sinal é deve ser propagado do LSB<sup>1</sup> até o MSB<sup>2</sup> ou o contrário?

**A ser entregue:** todos os arquivos de esquemático utilizados (.bdf), e arquivos de forma de onda usados para validar o projeto (.vwf).

### 1.2

Outra forma de projetar é utilizada quando já se dispõe de somadores/subtratores (como, por exemplo em um processador). Normalmente, no projeto do comparador na organização de um processador, ao invés de se projetar um circuito específico para executar a tarefa, reutiliza-se o circuito somador/subtrator. E, adicionando um circuito de lógica muito simples, são obtidos os três sinais ( $A < B$ ,  $A = B$  e  $A > B$ ) que indicam o resultado da comparação entre dois **números inteiros quaisquer**.

**Projete e simule** (com todas as possíveis entradas) um comparador de 4 bits, que tem como entrada duas palavras de 4 bits  $A(3:0)$  e  $B(3:0)$  e como saída 3 bits, *gt*, *lt*

---

<sup>1</sup> Least significant bit

<sup>2</sup> Most significant bit

e  $eq$ , que significam respectivamente  $A > B$ ,  $A < B$  e  $A = B$ . Você pode usar o diagrama de blocos da Figura 1 como auxílio.

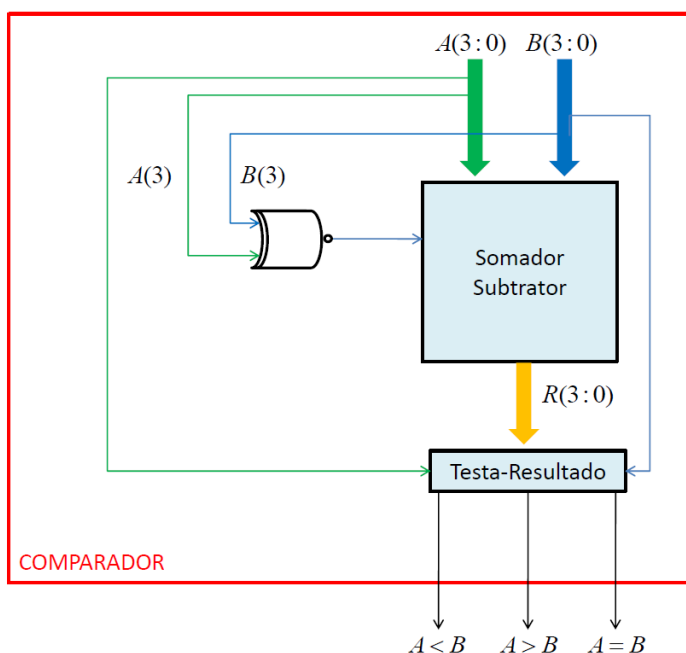


Figura 1: Diagrama de blocos do comparador da Questão 1.2.

**A ser entregue:** todos os arquivos de esquemático utilizados (.bdf), e arquivos de forma de onda usados para validar o projeto (.vwf).

## Q2.

Uma unidade aritmética lógica (*arithmetic-logic unit*) ALU é um módulo capaz de realizar um conjunto de funções aritméticas e lógicas. Para isto, uma ALU tem vetores de entrada/saída de dados, bem como entradas e saídas de controle. A Figura 2 mostra uma possível especificação de uma ALU de 4 bits.

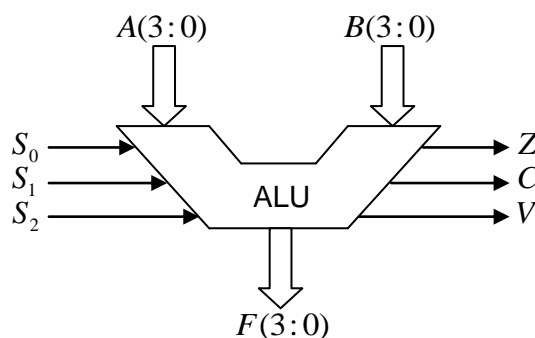
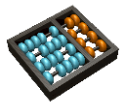


Figura 2: ALU de 4 bits.



A ALU da Figura 2 tem entradas ( $A$  e  $B$ ) de 4 bits,  $S_0$ ,  $S_1$  e  $S_2$  são entradas para selecionar a operação. A saída  $F$  de 4 bits mostra o resultado da operação.  $Z$  é 1 se o resultado da operação for zero e 0 caso contrário.  $C$  é 1 se houver um *carry-out* em operações de soma ou subtração.  $V$  é 1 se houver *overflow*.

Os valores das entradas e suas respectivas operações são mostrados na Tabela 1:

Tabela 1: Operações da ALU

$S_0$	$S_1$	$S_2$	Operação ( $F$ )
0	0	0	$A + B$
0	0	1	$A - B$
0	1	0	$A + 1$
0	1	1	$A - 1$
1	0	0	$\bar{A}$
1	0	1	$A \text{ and } B$
1	1	0	$A \text{ or } B$
1	1	1	$A \oplus B$

**Projete** a ALU especificada acima e **simule** sua operação de modo a validar seu funcionamento.

**A ser entregue:** todos os arquivos de esquemático utilizados (.bdf), e arquivos de forma de onda usados para validar o projeto (.vwf).

**Observação:** Preste bem a atenção nos bits de controle  $S_0$ ,  $S_1$  e  $S_2$ , eles foram selecionados para simplificar a seleção das operações, veja como na Figura 3 que apresenta o diagrama de blocos da ALU. Você pode utilizar componentes já projetados anteriormente (por exemplo o somador/subtrator da LE4).

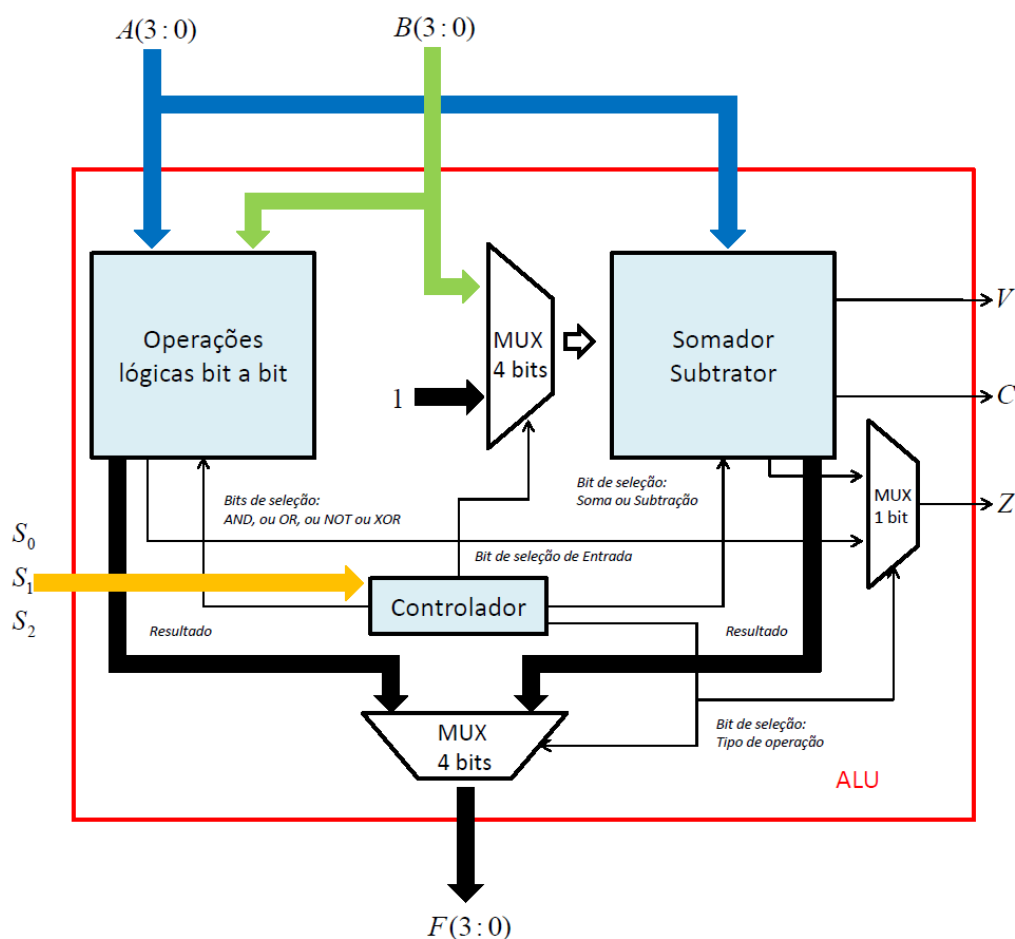


Figura 3: Diagrama de blocos da ALU da Q2.

### Q3.

A central telefônica da Figura 4 funciona para apenas quatro assinantes  $A(3:0)$ , com as seguintes características:

É possível manter apenas uma conversação por vez.

O valor de Source (Src) define qual assinante está originando a chamada.

O valor Destination (Dst) define quem recebe a chamada.

Os assinantes não envolvidos na conversação devem receber saída zero.

**Projete e simule** esta central telefônica como especificado. Utilize **multiplexador(es)**, **decodificador(es)** e portas lógicas convenientes para o projeto.

**A ser entregue:** todos os arquivos de esquemático utilizados (.bdf), e arquivos de forma de onda usados para validar o projeto (.vwf).

**Nota:** Recomenda-se que o projeto utilize componentes existentes de multiplexadores e decodificadores. O aluno pode projetar esses componentes e gerar seus símbolos, ou, se preferir, utilizar componentes da série 74xxx. Os componentes podem ser instanciados no Quartus na biblioteca Others > MaxPlus2. Observem que os componentes utilizados até agora (and2, or3, input, etc) estavam na biblioteca Primitives>Logic. Para descobrir qual componente utilizar e qual seu funcionamento

---

procurem o datasheet (Google datasheet decoder mux 74 series) em sites como por exemplo [http://www.hobbyprojects.com/datasheets/logic\\_74xxx\\_series\\_datasheets.html](http://www.hobbyprojects.com/datasheets/logic_74xxx_series_datasheets.html)

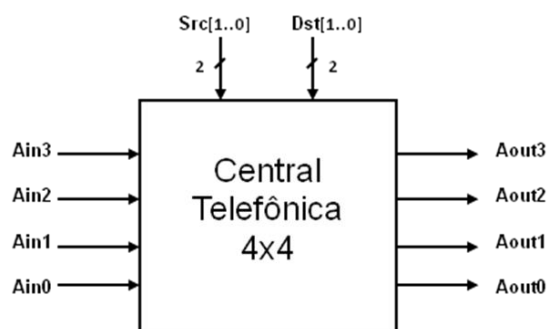


Figura 4: Central telefônica.

#### Q4. (BV 6.7)

Considere a função  $f = \bar{w}_2 + \bar{w}_1\bar{w}_3 + w_1w_3$ . Utilizando a expansão de Shannon, desenhe um circuito para f, com base em multiplexadores

**A ser entregue:** solução do exercício em papel