



INSTITUTO METROPOLE DIGITAL
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA

IANCO SOARES OLIVEIRA

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN

RELATÓRIO (CIRCUITOS LÓGICOS)

NATAL
2022

INTRODUÇÃO

O objetivo deste relatório de trabalho é explicar as etapas de implementação de um circuito (vou chamar de Circuito Operador) que tem o seguinte comportamento:

O circuito tem 2 entradas (X,Y) com números em C-2 de 4 bits e 4 saídas (A,B,C,D), cada uma conectada a um display de 7 segmentos.

O circuito implementa as seguintes operações:

- 1) se $X \geq 0$ mostrar o valor do módulo de X em decimal no display da saída A
- 2) se $X < 0$ mostrar o valor do módulo de X em decimal no display da saída B
- 3) se $Y \geq 0$ mostrar o valor do módulo de Y em decimal no display da saída C
- 4) se $Y < 0$ mostrar o valor do módulo de Y em decimal no display da saída D

DESCRIÇÃO

A partir da descrição do comportamento do circuito percebe-se que ele faz quatro operações muito parecida, cada uma das operações faz quatro ações, são elas: (i) compara um número (de 4 bits em complemento de 2) com zero, (ii) incrementa ou decrementa um número, (iii) calcula o valor absoluto e (iv) decodifica um número para ser exibido em um display de 7 segmentos.

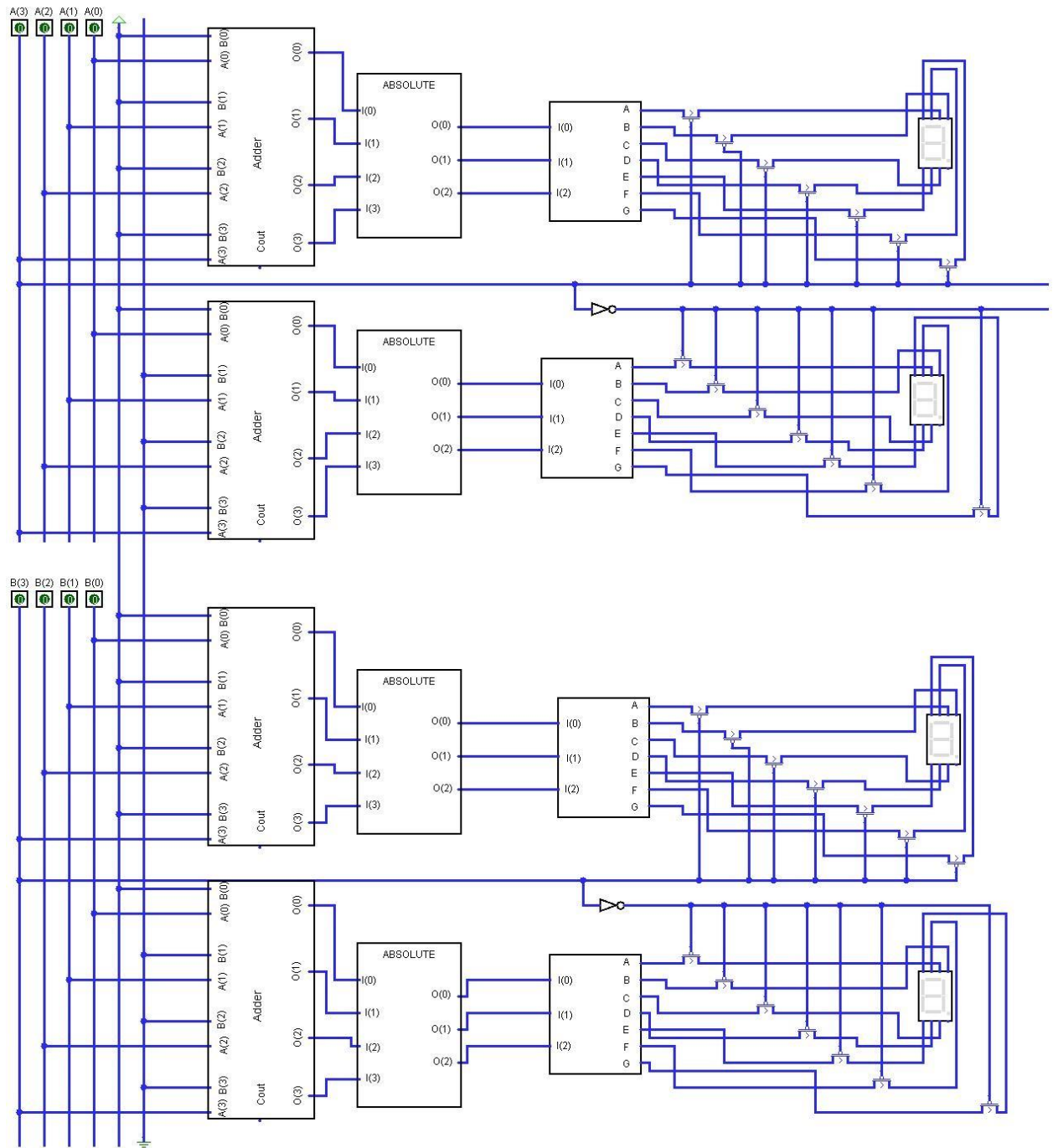
A ação (i) dispensa um componente específico, como um número em complemento de 2 o primeiro bit informa se ele é positivo ou negativo, então basta utilizar o bit mais significativo de cada uma das entradas para executar essa ação.

A ação (ii) pode ser completamente contemplada com a utilização de um somador de 4 bits cujo parâmetro é $0001_{2^c} = 1_{10}$ caso queria-se incrementar e $1111_{2^c} = -1_{10}$.

A ação (iii) pode ser realizada de diversas formas, a maneira mais intuitiva seria converter o número em complemento de 2 para uma representação padrão de inteiros em binário. Para isso seria necessário utilizar um somador e um inversor de bits. Outra forma seria criar um tipo de decodificador, como um número binário de 4 bits pode representar até 16 valores, é simples mapear cada um dos valores possíveis para os seus valores absolutos, e como existem apenas metade dos valores absolutos, então o número de saída pode ser representado por 3 bits. A segunda estratégia foi adotada nessa implementação

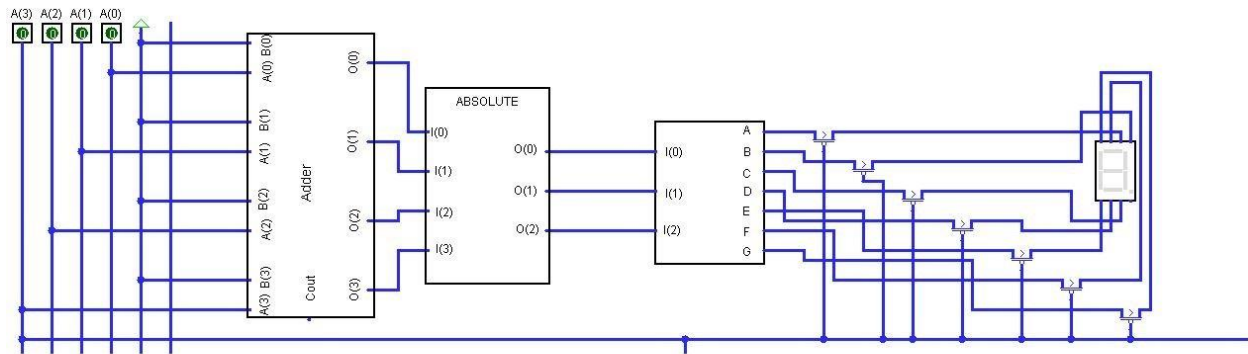
A ação (iv) é um decodificador, que mapeia as possibilidades de entrada para valores que podem ser exibidos em displays de 7 segmentos.

Então vamos ver como ficou o formato geral do circuito:



CIRCUITO OPERADOR

Perceba que há quatro repetições da mesma estrutura, isso se deve ao fato das quatro operações necessárias ao circuito usarem os mesmos componentes nas mesmas ordens, portanto o entendimento da implementação de uma das operações é suficiente para o entendimento de todo o circuito. Então, para melhor visualização, darei na primeira operação.

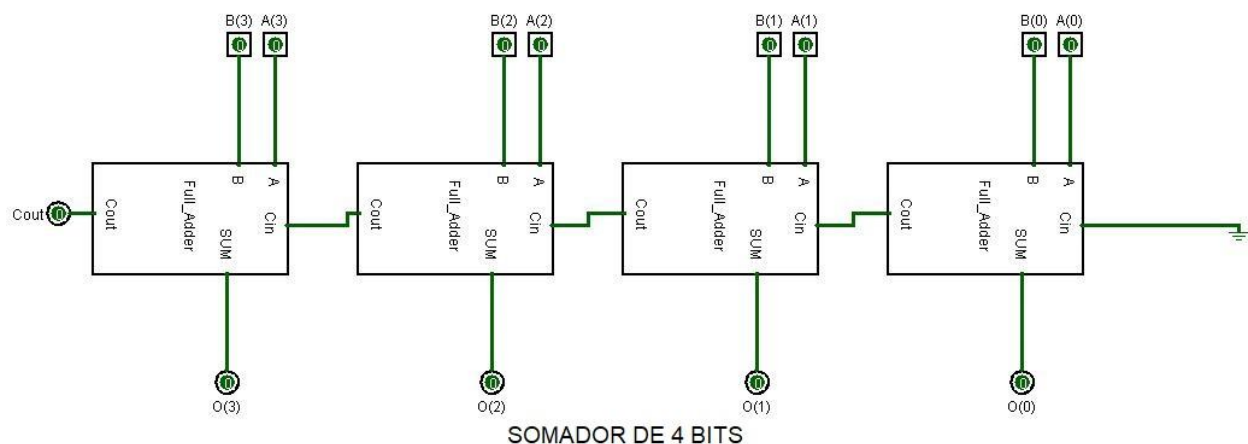


CIRCUITO OPEDADOR (primeira operação)

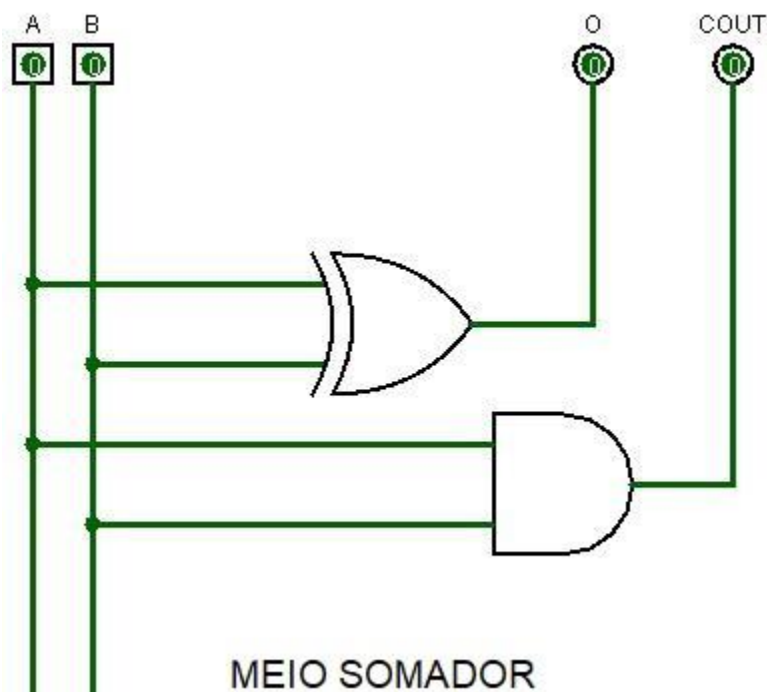
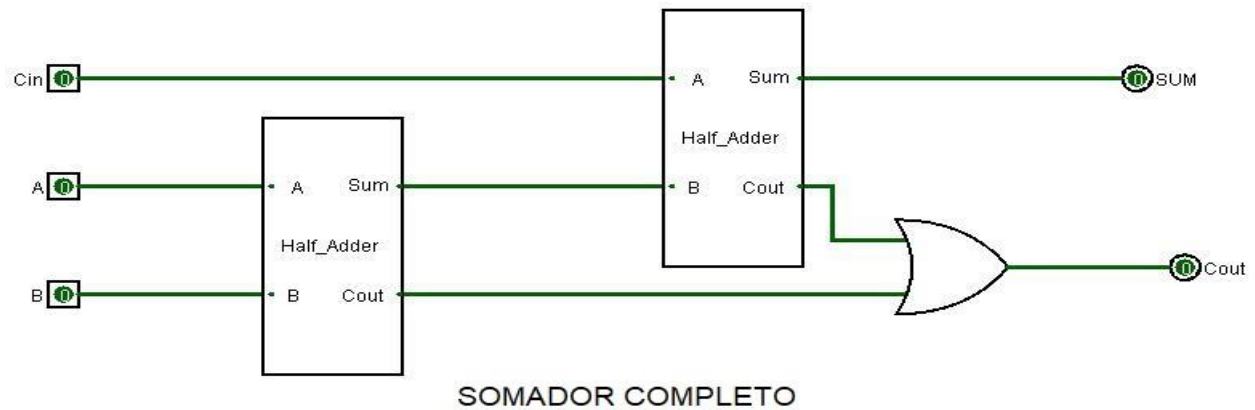
DIAGRAMAS

A ação (i) aqui é analisada por 7 transistores que recebem como entrada cada um dos sinais a serem enviados ao display de 7 segmentos e o controle é o bit mais significativo da entrada do Circuito Operador. Esses transistores poderiam ser trocados por um multiplexador 2x1 com uma das entradas sendo um vetor de 7 bits, a segunda entrada aterrada e o seletor sendo o primeiro bit da entrada do Circuito Operador, a escolha por transistores neste diagrama se deu por ser mais simples utilizar esse método na ferramenta que foi utilizada para criar esse diagrama.

A ação (ii) é executada pelo *addar*, segue o diagrama de blocos desse componente:

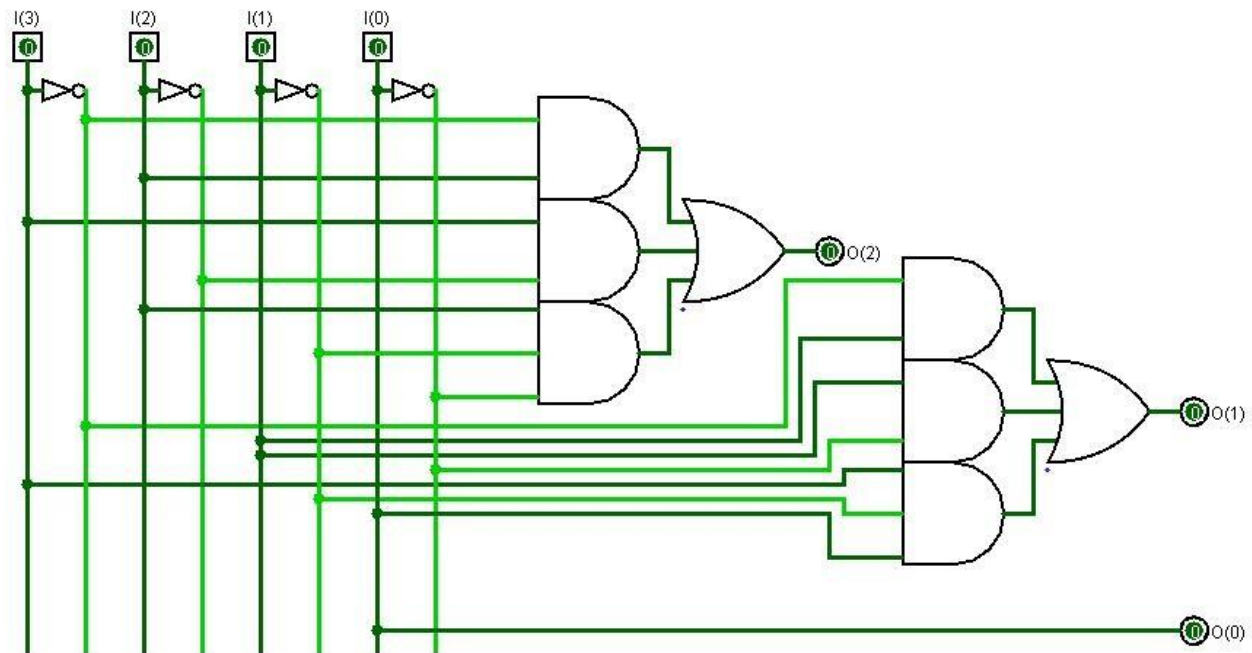


SOMADOR DE 4 BITS



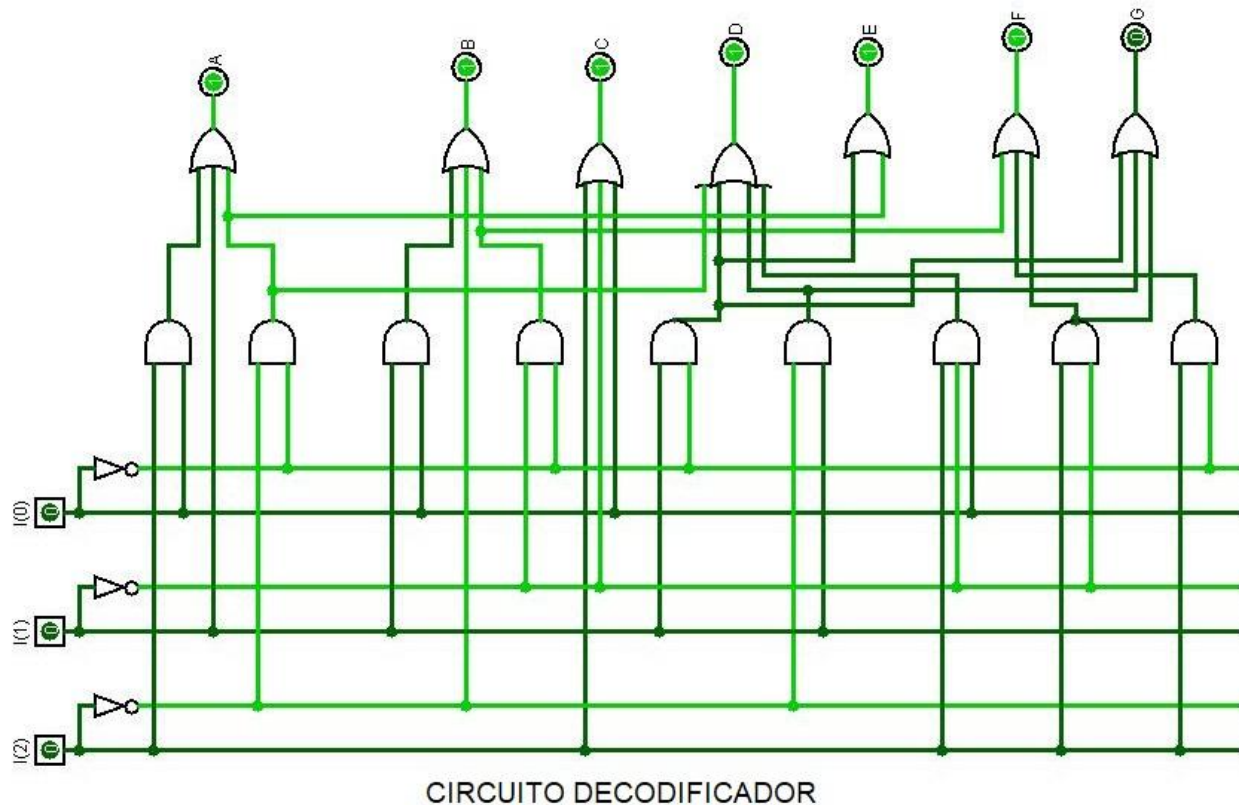
Esse circuito é uma implementação clássica de um somador de 4 bits implementado com quatro somadores completos que, por sua vez, é implementado com dois meio somadores. Basta passar 1_{10} ou -1_{10} para incrementar ou decrementar um valor.

Em seguida temos a ação (iii) aqui representada pelo circuito de valor absoluto. Que nada mais é que um decodificador que recebe 4 bits como entrada e mapeia cada um dos valores para uma saída de 3 bits que representa o valor absoluto do valor da entrada. Segue o circuito:



CIRCUITO DE VALOR ABSOLUTO

Por fim, a última ação, a ação (iv) é facilmente realizada por um circuito decodificador para um display de 7 segmentos. Esse tipo de circuito, assim como o anterior, mapeia cada uma das entradas para cada uma das saídas. Veja, como esse decodificador se preocupa apenas com valores de 0_{10} a 7_{10} então ele não implementa todas as lógicas de um circuito decodificador clássico, mas ainda assim é muito parecido. Segue:



FERRAMENTAS

Cada um desses diagramas foi produzido com auxílio do aplicativo [logsim-evolution](#) que é um projeto em java e Open Source e pode facilmente ser baixado e utilizado.

A implementação foi escrita em VHDL de forma estrutural e tenta reproduzir exatamente o que foi desenhado nos diagramas, entretanto existem pequenas diferenças entre o circuito que é "compilado" pelo [Quartus Prime](#) e o circuito dos diagramas. Isso se deve ao fato de ser mais simples descrever alguns processos em VHDL do que na ferramenta logsim-evolution. Entretanto, salvo essas pequenas diferenças, os diagramas gerados pelo Quartus Prime e os aqui apresentados serão bastante parecidos.

Por fim, esse projeto apresenta um outro código VHDL (veja o [link](#)) que representa um circuito de testbench. Esse teste pode ser utilizado em uma ferramenta como o [modelsim](#) para gerar arquivos de forma de onda para analisar o comportamento do circuito. Entretanto o repositório do projeto já apresenta um diagrama (veja o [link](#)).