

Projeto de circuito

MAC0329 – Álgebra booleana e aplicações (DCC / IME-USP — 2020)

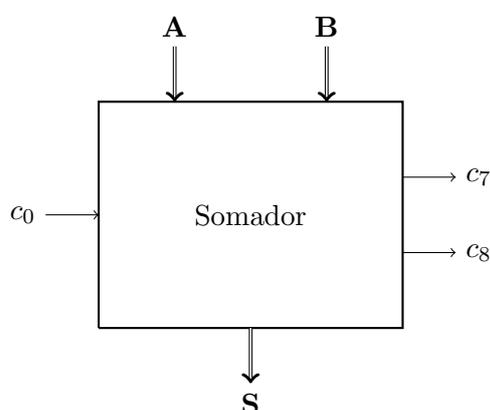
– Todas as etapas do projeto deverão ser feitas na plataforma <https://circuitverse.org>

Parte 1: Circuito Somador – entrega na plataforma CircuitVerse, até 29/03

Nesta primeira etapa do projeto (Parte 1), o objetivo é a implementação de um circuito somador de 8 *bits*, e sua utilização para realizar as operações de adição e subtração com números de 8 *bits*.

1 Especificação de um somador de 8 *bits*

Em nosso projeto, o somador de 8 bits terá a seguinte configuração



Entradas:

$\mathbf{A} = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ (operando 1)

$\mathbf{B} = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ (operando 2)

c_0 (*carry* na coluna 0)

Saídas:

$\mathbf{S} = \mathbf{A} + \mathbf{B}$ (8 *bits* da soma)

c_7 (*carry* na coluna 7)

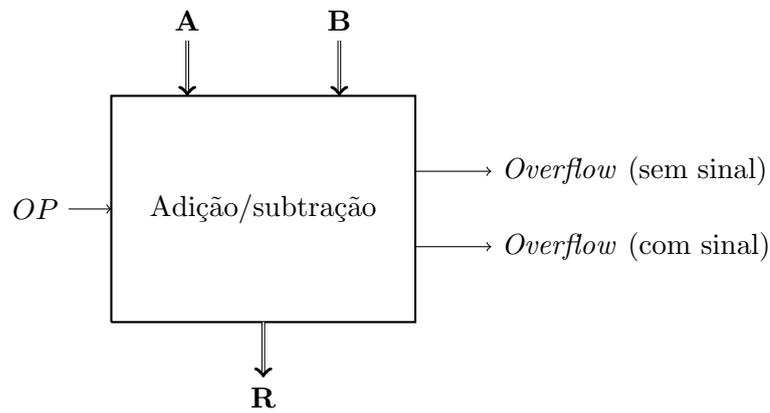
c_8 (*carry* na coluna 8)

2 Modularização

Modularização relaciona-se com organização. No caso de circuitos lógicos, um circuito complexo pode ser decomposto em termos de combinação de subcircuitos. Cada subcircuito pode ser visto como uma “caixa-preta” e deve ter uma interface e funcionalidade bem definidas. Com isso, conseguimos organizar um circuito complexo de forma compacta, o que facilita o seu entendimento, mesmo que não saibamos exatamente como cada subcircuito está implementado. Além disso, um subcircuito pode ser utilizado em diversas partes de um circuito maior, evitando repetições.

Você deverá organizar o circuito do EP1 em três circuitos/subcircuitos:

1. **main** : o circuito principal, que servirá para testar o somador. Aqui espera-se um circuito com a seguinte interface:

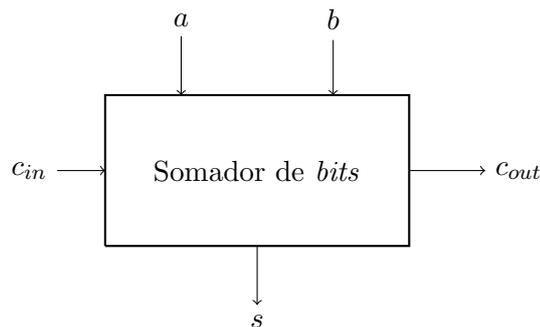


Neste diagrama,

- **A** e **B** são os operandos, ambos de 8 *bits*;
- **OP** é um *bit* para indicar o tipo de operação: se 0, deve ser calculada a adição $\mathbf{A + B}$ e se 1 deve ser calculada a subtração $\mathbf{A - B}$;
- **R** é o resultado da operação (8 *bits*);
- “*overflow sem sinal*” é um *bit*: se 1, indica que a operação calculada, interpretando-se os números como sem sinal, resultou em *overflow*;
- “*overflow com sinal*” é um *bit*: se 1, indica que a operação calculada, interpretando-se os números como com sinal (na notação complemento de dois), resultou em *overflow*.

O circuito principal deve utilizar o circuito **somador**, descrito no próximo item.

2. **somador**: este deve ser o circuito somador descrito na seção 1 acima e ele deve ser construído usando o subcircuito **somabits** especificado a seguir.
3. **somabits**: este deve ser o circuito somador de *bits*. Ele recebe três *bits*, a , b e c_{in} , e devolve dois *bits*, s e c_{out} , conforme mostrado no diagrama a seguir (veja também o capítulo 3 das notas de aula).



Observação: Embora aqui estejamos falando de modularização de circuitos lógicos, esse conceito estará bastante presente na especificação e desenvolvimento de softwares em geral. Em MAC0110, em breve vocês aprenderão “funções”, que correspondem a um dos elementos modulares na organização do código de um programa.

3 O que fazer e entregar

O *CircuitVerse* permite a criação de vários subcircuitos, que podem ser atrelados a um único projeto.

Vocês deverão criar um projeto EP1 contendo os três circuitos descritos acima. Esta é a primeira vez que usamos o *CircuitVerse*, então iremos aos poucos descobrindo como fazer as coisas. Postem dúvidas sobre a utilização do *CircuitVerse* no fórum de discussões no Moodle.

A ordem natural para o EP1 é fazer o circuito seguindo a estratégia *bottom-up*, isto é, começa-se com o subcircuito mais básico (*somabits*), em seguida faz-se o *somador*, e finalmente o *main*. Cada um deles deve ser testado individualmente, antes de ser empregado em um circuito de nível superior.

Os seguintes elementos devem ser suficientes para a construção dos circuitos acima:

- entrada e saída
- portas lógicas
- *buffers* e inversores (controlados ou não)
- *splitters*

4 Critérios de avaliação

Serão avaliados os seguintes aspectos:

- corretude dos circuitos
- aderência à organização proposta, com uma clara identificação (nome consistente) dos pinos de entrada e saída em cada circuito
- cumprimento do prazo