

21b - AV2

Avaliação 2 - Elementos de Sistemas

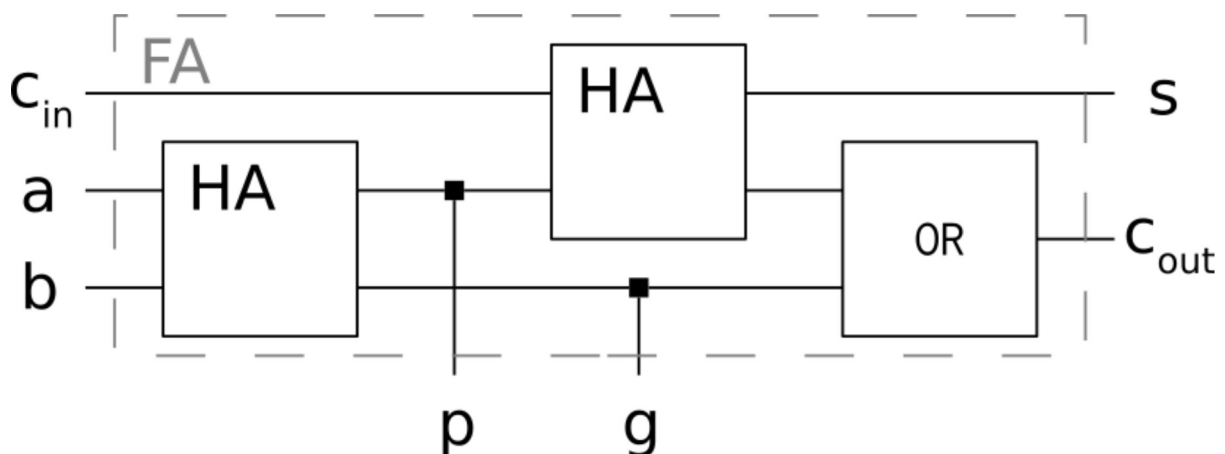
AV	Pontos HW	Pontos SW
Prática	35	5

- **Trabalhar sozinho.**
- **100 min total.**
- Ficar conectado no canal geral (para ouvir instruções)
- **Commit a cada 15 minutos**
- Lembre de dar push ao final

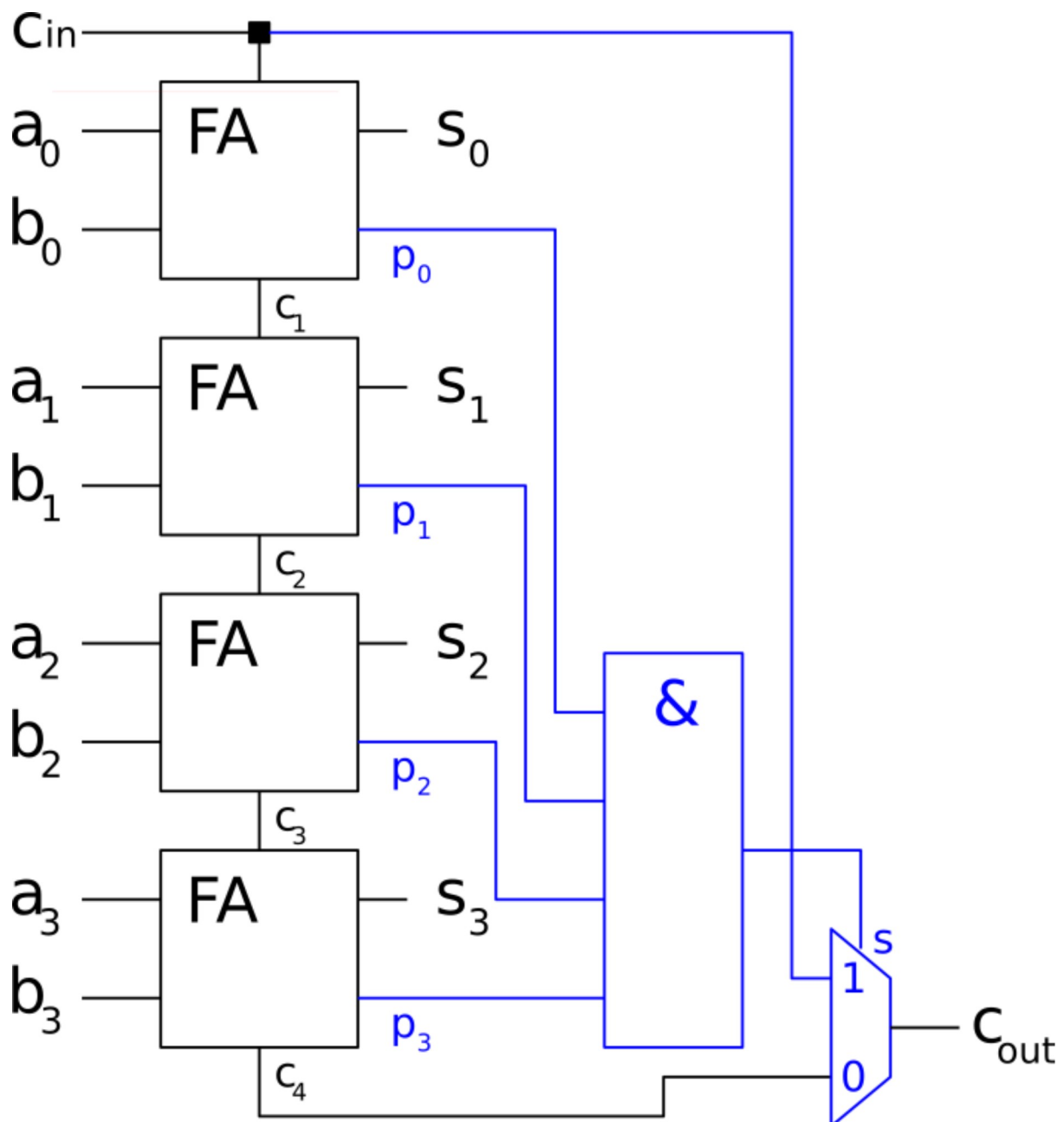
Para testar basta descomentar o módulo que deseja validar no arquivo `config_testes.txt` e executar o comando `python3 ./testeHW.py`.

1. Novo adder

Existem várias topologias de somadores (adder) cada um com uma vantagem e desvantagem, o proposto a seguir tem o intuito de minimizar o tempo de cálculo da soma, para isso utiliza um FullAdder modificado com a adição de dois sinais de saída 'g' e 'p':



Com este FA é possível implementar o Adder com skip, conforme diagrama a seguir:



1a. Implementando

Arquivo	pnts
src/FullAdder_skip.vhd	5 HW
src/add4_skip.vhd	5 HW

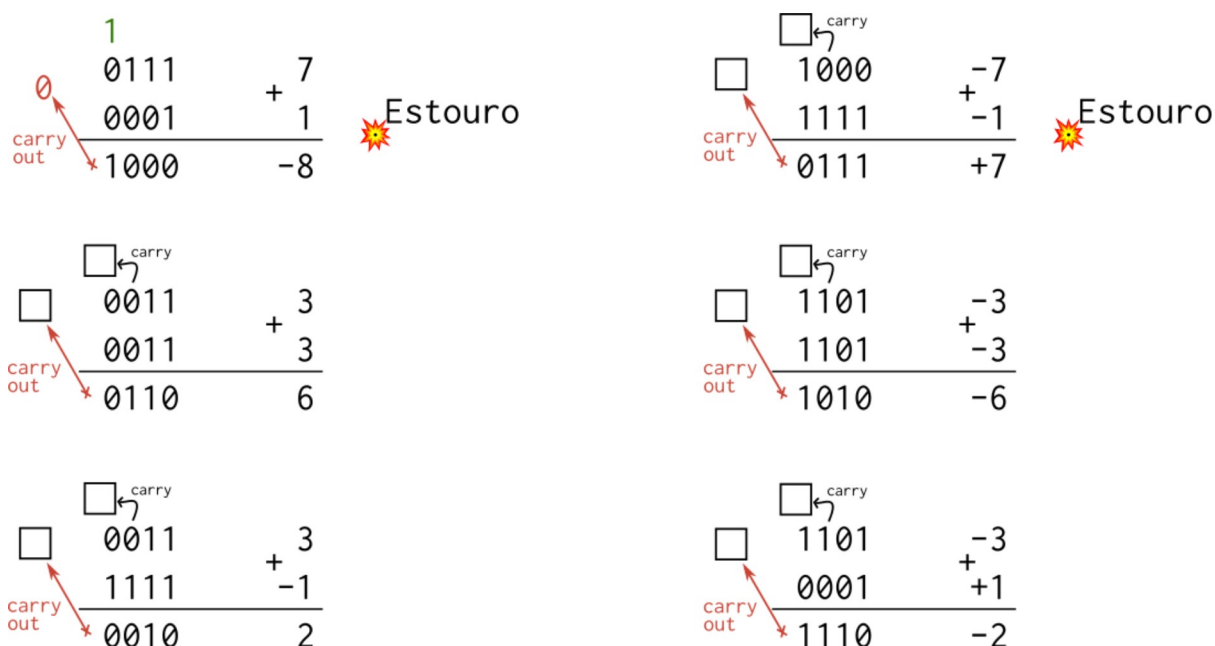
Usando o `HalfAdder.vhd` e o `Mux2Way.vhd` implemente o `FullAdder_skip.vhd` e e na sequência o `Add4_skip.vhd` conforme diagramas anteriores..

Lembre de descomentar o módulo no arquivo `config_testes.txt` e testar com `./testeHW.py`

2. (10 HW/ 0 SW) Adder Overflow

Arquivo	pnts
src/add4.vhd	10 (HW)

Um problema da soma binária de vetores sinalizados (que podem ser positivos ou negativos) é que pode acontecer um estouro na soma, este estouro acontece quando somamos dois valores e o resultado não cabe na quantidade de bits reservado para armazenar o resultado. Note nos exemplos a seguir (4 bits) que a indicação de estouro é diferente do carry out.



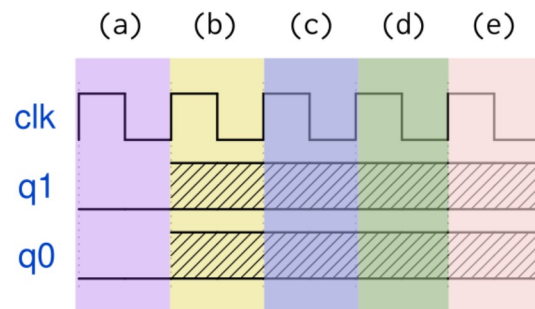
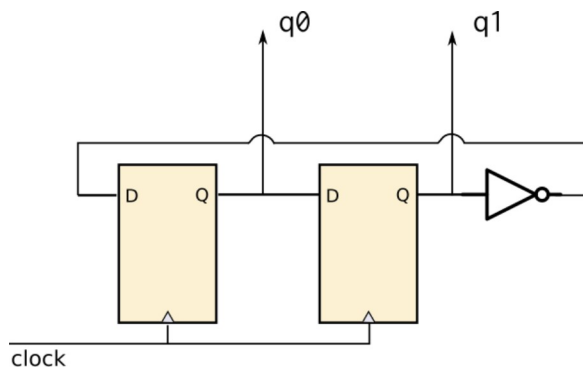
Agora modifique o somador de 4 bits (`add4.vhd`) para que o sinal de estouro represente quando aconteceu um estouro de verdade.

Lembre de descomentar o módulo no arquivo `config_testes.txt` e testar com `./testeHW.py`

3. Circuito misterioso ?

Arquivo	pnts
src/q3.vhd	5 (HW)

Analise o circuito a seguir e responda no arquivo `src/q3.vhd` qual o valor de `q1` e `q0` nos instantes (b), (c), (d), (e). Você deve responder no arquivo VHDL, use como exemplo a resposta do instante (a).



Essa questão não tem teste!

4. Paridade

Arquivo	pnts
src/paridade.vhd	5 (HW)

Vocês devem desenvolver em VHDL um componente que identifica a paridade de um vetor de 4 bits, indicando quando a soma de bits 1 é ímpar ou par (saídas), conforme o exemplos a seguir:

- Não pode fazer uso de with nem when.

```
0101 --> par    = 1
      \-> impar  = 0

1101 --> par    = 0
      \-> impar  = 1
```

Lembre de descomentar o módulo no arquivo `config_testes.txt` e testar com `./testeHW.py`

5. ULA

Arquivo	pnts
src/q5.vhd	5 (HW) / 5 (SW)

Vamos falar um pouco da ULA desenvolvida no projeto C-ULA, no arquivo `q5.txt` responda:

- Como um programa de alto nível usa a ULA e consegue executar operações não existentes nela, como Multiplicação ou Divisão?

