3. Implemente um **módulo somador de 4bits**, sem entrada de carry-in, nem saída de carry-out. Escolha o nível de abstração que mais lhe convém. *Dica: assista o video*.

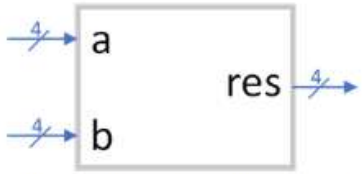

Diagrama de blocos do somador de 4bits

4. Implemente um testbench simples, por meio de delays, para validar um cenário de 5 somas distintas utilizando seu **módulo somador de 4bits**. Simule-o no EDA Playground e cheque se os resultados estão de acordo com o previsto. Chame o professor para Validar seu progresso até esse ponto.

5. Implemente um **módulo MUX 2x1 de 4bits**. *Dica: assista o video*.

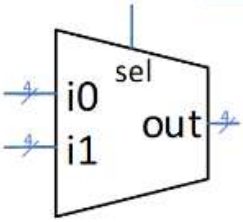

Diagrama de blocos do MUX 2x1 de 4bits

6. Implemente um testbench para validar um conjunto representativo de entradas e saídas do seu **módulo MUX 2x1 de 4bits**. Simule-o no EDA Playground e cheque se os resultados estão de acordo com o previsto. Chame o professor para Validar seu progresso até esse ponto.

7. Utilize os módulos de **somador** e **MUX 2x1** para construir um terceiro módulo que possibilite operar entre duas entradas **A** e **B** ou entre **A** e uma constante **C**. A seleção ente os dois modos de operação, deve ser realizada por uma entrada **S**.

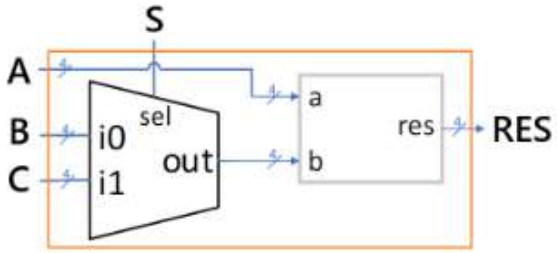

Diagrama de blocos da montagem com o MUX e o somador

8. Implemente um testbench para validar a sua montagem do item 7. Simule-o no EDA Playground e cheque se os resultados estão de acordo com o previsto. Chame o professor para Validar seu progresso até esse ponto.

Após o professor conferir seus testes, compacte todos os arquivos em um **.zip** e submeta-o no **SIGAA**

Desafio (Valendo +0,1 na média geral)

1. Modifique seu somador para possibilitar a execução de 4 operações (ULA): Soma, Subtração, deslocamento para direita e para esquerda. Além da saída do resultado, inclua um flag de status para indicar overflow.

Sel	Res
00	A + B
01	A – B
10	A >> B
11	A << B

2. Implemente um testbench para validar os cenários mais significativos do seu novo módulo de ULA.