

Projeto de circuito 2 – entrega no PACA, até 11/05

A **ULA** (unidade lógico-aritmética) é a parte do processador responsável pelas operações lógicas e pelas operações aritméticas. O objetivo deste projeto é criar parte do circuito de uma ULA, conforme detalhado mais adiante. Note que a ULA que especificamos aqui é apenas um modelo simples.

Para o desenvolvimento do projeto deve ser usado o software Logisim (<http://www.cburch.com/logisim/>).

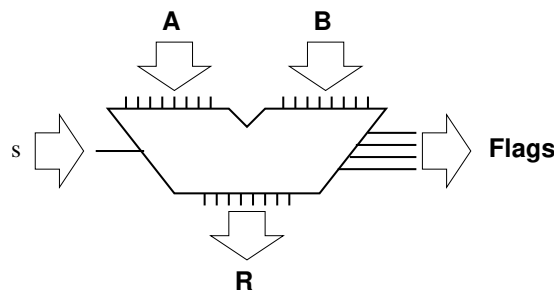
O projeto pode ser desenvolvido em grupos com até 3 membros. O desenvolvimento deste e demais projetos em grupo é fortemente recomendado (se possível, mantendo os grupos). Todos os membros devem participar ativamente das discussões e do planejamento e ter ciência sobre os detalhes do projeto desenvolvido pelo grupo.

A entrega do projeto será via PACA. Devem ser entregues o arquivo `.circ`, criado com o Logisim, e contendo o circuito da ULA, mais um documento (pode ser `txt` simples) contendo o nome dos membros do grupo e uma explicação sucinta e clara de como está organizado o circuito. Os dois arquivos devem ser empacotados em um único arquivo (`.zip` ou `.tar.gz` ou `tgz`).

Postem suas dúvidas ou descobertas no Fórum da disciplina.

Detalhamento

Deste projeto em diante iremos considerar palavras de 8 *bits*. Vamos também supor que todos os números estão na notação complemento de dois. Assim, os números que podem ser representados em uma palavra variam de -128 a 127. O esquema da ULA a ser implementada é mostrado na figura a seguir.



As entradas A e B na parte superior ($A = a_7 a_6 \dots a_1 a_0$ e $B = b_7 b_6 \dots b_1 b_0$) correspondem aos dois números a serem operados, enquanto a saída R na parte inferior ($R = r_7 r_6 \dots r_1 r_0$) corresponde ao resultado da operação (quando for o caso). Na lateral esquerda temos um pino seletor s que serve para indicar a operação aritmética a ser executada, e na lateral direita temos alguns *flags* de saída.

O seletor deve funcionar da seguinte forma:

s	Operação a ser executada
0	Adição ($A + B$)
1	Subtração ($A - B$)

Já as *flags* de saída consistem de 4 *bits*, $o_3 o_2 o_1 o_0$, sendo o_0 a primeira e o_3 a última de cima para baixo. As *flags* indicam os seguintes estados:

$$\begin{aligned}
o_0 = 1 &\iff \text{overflow na operação aritmética} \\
o_1 = 1 &\iff A > B \\
o_2 = 1 &\iff A = B \\
o_3 = 1 &\iff A < B
\end{aligned}$$

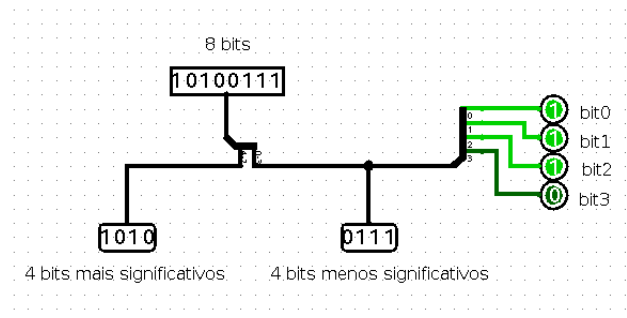
A ULA deverá funcionar de tal forma que sempre que ocorre alguma alteração em alguma de suas entradas (no caso, em A , B ou s) as saídas (R e as quatro *flags*) sejam alteradas imediatamente em seguida, de acordo com as novas entradas.

Observações e dicas:

- O Logisim possui alguns módulos prontos. Neste projeto, porém, é esperado que todo o circuito seja implementado sem o uso de módulos prontos do Logisim, com exceção do multiplexador (seletor de dados).
- Planeje a organização do circuito antes de começar a desenhá-lo no Logisim. Uma boa prática é organizar o circuito em módulos (subcircuitos). Assim, à medida que o circuito é incrementado, tornando-se cada vez mais complexo, os subcircuitos podem ficar “encapsulados” em uma caixinha, permitindo uma organização hierárquica, mais fácil de ser entendido. O circuito principal deve ser nomeado *main*.

Além da organização hierárquica, pense também o leiaute do desenho em si.

- Em certas partes do circuito pode ser conveniente utilizar *bits* de dados “largos”, assim como os *splitters*. A figura a seguir ilustra um pino de entrada com 8 *bits* e o uso de *splitters* para separar (sub)grupos de *bits*.



- Verifique se a sua ULA calcula corretamente $-128 - (-128)$

