

3 - Unidade Lógica e Aritmética

EA773 - 2s2015

Tiago Tavares e Marcos Couvre

Introdução

Uma Unidade Lógica e Aritmética (ULA) é um dispositivo que realiza operações lógicas e aritméticas sobre números representados em circuitos lógicos. Tipicamente, uma ULA recebe dois operandos como entrada, e uma entrada auxiliar de controle permite especificar qual operação deverá ser realizada. Por esse motivo, a construção de uma ULA se baseia em dois fundamentos principais: o *controle de fluxo de dados* e a construção de *circuitos que implementam operações*.

Controle de fluxo

Controlar o fluxo de dados, em um circuito lógico, significa modificar, como parte do funcionamento do circuito, as conexões entre entradas e saídas de componentes. Há dois casos que são tipicamente abordados na literatura: a *multiplexação* (várias entradas compartilham a mesma saída) e a *demultiplexação* (uma entrada é ligada a apenas uma dentre várias saídas possíveis). Uma das formas de implementar multiplexadores e demultiplexadores envolve o uso de lógica de três estados (tri-state).

Um componente tri-state tem uma entrada adicional de controle, na qual não há fluxo de dados. Essa entrada pode ser utilizada para desabilitar o componente. Nesse caso, a saída do componente assume um estado de *alta impedância* e passa a se comportar como se tivesse sido desconectada do circuito.

Assim, um multiplexador pode ser construído ligando várias entradas a uma só saída usando buffers tri-state. Uma entrada de controle recebe o número da entrada de dados que deve ser habilitada, e os buffers correspondentes a todas as outras entradas são configurados para estado de alta impedância. Esse projeto, mostrado na Figura 1, é bastante conhecido.

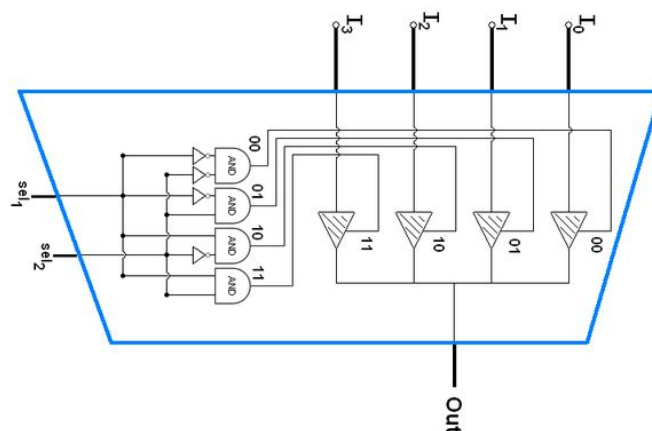


Figura 1: Esquema lógico de um multiplexador, retirado da Wikimedia Commons.

Um demultiplexador pode ser construído usando um raciocínio semelhante. Desta vez, porém, a entrada de controle definirá a qual das várias saídas a entrada deverá ser conectada. Este projeto também é bastante conhecido e pode ser implementado facilmente.

Operações aritméticas: somador

Um somador é um circuito que recebe dois números e produz, na saída, a soma desses dois números. O projeto desse circuito depende de como o número será representado - é possível, por exemplo, construir somadores analógicos se um número for representado por um nível de tensão correspondente. Em circuitos lógicos, é uma prática comum representar números relacionando cada bit a um dígito desse número em base 2.

Uma arquitetura possível para um somador parte da soma de um bit. Um operando de 1 bit pode assumir os valores 0 ou 1. Como sabemos, $0+0=0$, $0+1=1$, $1+0=1$ e $1+1=2$, e, portanto, um somador de 1 bit pode ser descrito pelas regras mostradas na Tabela 1, considerando que Co é o bit mais significativo do resultado e Y é o bit menos significativo.

A	B	Y	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabela 1: Tabela-verdade de um somador de 1 bit.

O somador com operandos de 1 bit, portanto, precisa de 2 bits em sua saída para representar o resultado da operação, sendo que o bit mais significativo corresponde ao “vai um” (chamado de *Carry* na literatura de circuitos lógicos) de uma conta de soma. Assim, se quisermos adicionar um segundo bit aos operandos, será preciso somá-lo considerando o *carry*. Nesse caso, devemos considerar, além dos casos que deram origem à Tabela 1, que $1+0+0=1$, $1+0+1=2$, $1+1+0=2$ e $1+1+1=3$, e, portanto, o somador relativo ao segundo bit pode ser descrito pelas regras mostradas na Tabela 2.

A	B	Ci	Y	Co	A	B	Ci	Y	Co
0	0	0	0	0	0	0	1	1	0
0	1	0	1	0	0	1	1	0	1
1	0	0	1	0	1	0	1	0	1
1	1	0	0	1	1	1	1	1	1

Tabela 2: Tabela-verdade de um somador de 1 bit com carry.

O bit mais significativo do resultado gerado pela soma dos bits menos significativos é, no contexto da operação aritmética, um operando na soma do próximo bit. O resultado relacionado ao segundo bit menos significativo também gera um bit de carry, que pode ser propagado, e assim por diante, por um número arbitrário de bits. Essa estratégia de construção de um somador se chama *carry-ripple* (LIN, 2003).

Unidade Lógica e Aritmética (ULA)

Uma unidade lógica e aritmética é um dispositivo capaz de realizar operações lógicas (AND, OR, rotação, shift, etc.) e aritméticas (soma, subtração, etc.). Uma ULA recebe como entrada um conjunto de operadores e realiza sobre eles uma de suas operações, escolhida através de uma entrada de controle. Retorna, na saída, o resultado das operações e um conjunto de bits de *status*, comumente chamados de *flags*.

Dessa forma, o projeto de uma ULA depende da implementação de operadores lógicos e aritméticos e do controle de fluxo através deles, através de multiplexadores e demultiplexadores. A Figura 2 mostra um possível diagrama de blocos de uma ULA, mostrando como os bits de controle determinam qual operador receberá os operadores de entrada e fornecerá a saída. É importante ressaltar que esta não é a única forma de construir uma ULA e, provavelmente, não é a mais eficaz.

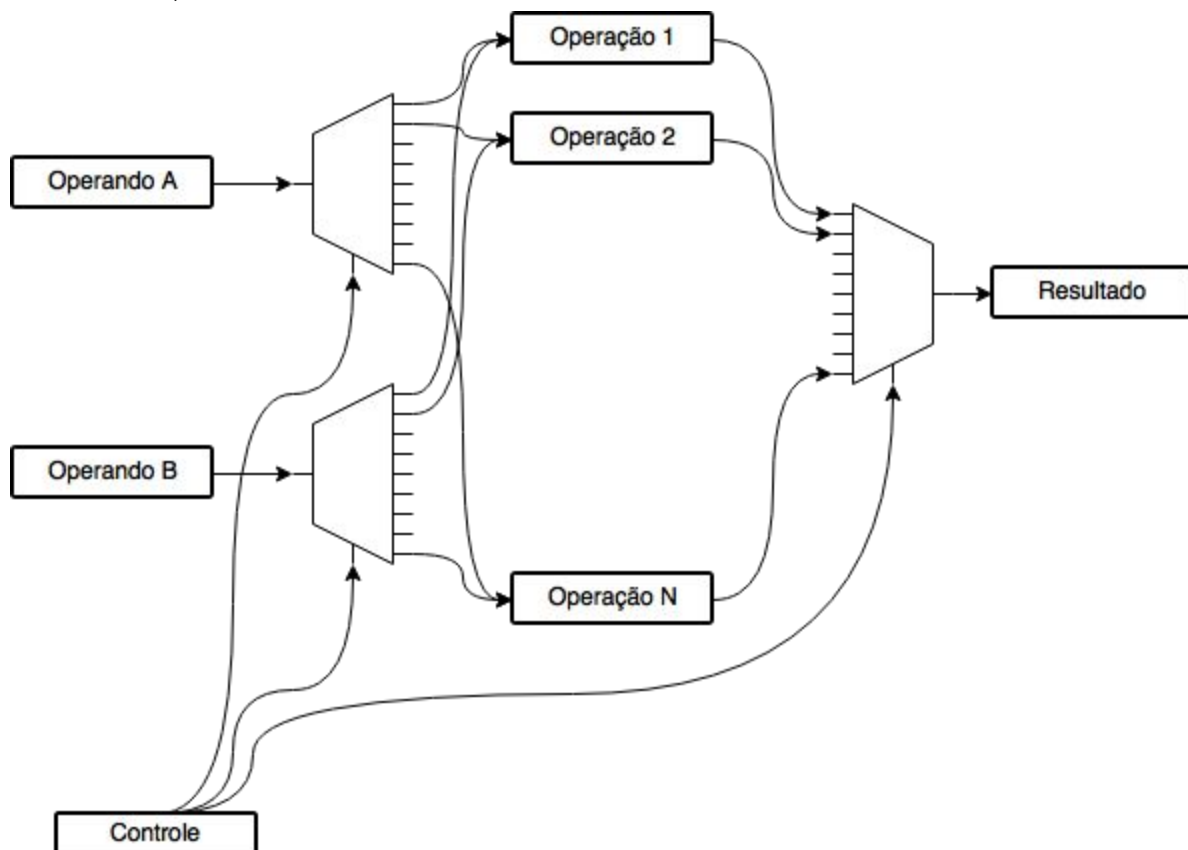


Figura 2: Diagrama de blocos de uma ULA.

As flags de status evidenciam informações sobre a operação que foi realizada. É normal que cada arquitetura de ULA forneça um conjunto diferente de flags de status, mas alguns deles se tornaram bastante usuais. Esses bits, e seus significados, são descritos na Tabela 3.

Bit	Significado	Condição
C	Carry-out	Houve <i>carry-out</i> no bit mais significativo do operador.
Z	Zero	O resultado é zero.

Tabela 3: Significados e condições relacionadas às flags de status

FPGA

FPGA é o acrônimo para Field-Programmable Gate Array. Trata-se de um dispositivo no qual é possível carregar instruções que definem um circuito lógico, e esse circuito será implementado no interior do circuito integrado através do re-arranjo de sinais lógicos. No laboratório, utilizaremos uma placa FPGA que, adicionalmente, traz LEDs e botoeiras que poderão ser usadas nos procedimentos de teste. Desta forma, os circuitos projetados no Quartus II podem receber entradas de chaves e acender LEDs (de forma semelhante à placa de teste do primeiro experimento). Para isso, existe um procedimento correto, e há tutoriais (Ting & Fazanaro, 2010) e manuais (ALTERA, 2006) disponíveis online explicando detalhadamente todos os passos desse procedimento.

Montagem

Nesta aula, montaremos uma pequena ULA. Para tal, utilizaremos o simulador Quartus II e a placa FPGA. O objetivo desta montagem é **desenvolver e testar uma ULA**. Para isso, o grupo deverá:

1. Projetar e montar um somador que recebe duas entradas de 4 bits,
2. Projetar e montar um circuito que realiza o XOR bit-a-bit de dois operadores de 4 bits,
3. Projetar e montar os multiplexadores e demultiplexadores necessários para realizar o controle de fluxo de dados da ULA,
4. Combinar esses circuitos de forma a montar um circuito que:
 - a. recebe como entrada dois inteiros de 4 bits e um sinal de controle que permite definir qual operação será realizada,
 - b. gera na saída um resultado de 4 bits e tres flags de status (C, Z)
 - c. permite ao usuário escolher se a operação realizada é uma soma ou um XOR bit-a-bit através de alguma chave na placa, definida pelo grupo
5. Carregar a ULA gerada na placa, associando botões às entradas e LEDs às saídas
6. Testar a ULA, gerando, no mínimo, casos de teste para o operador soma tais que os bits de status gerados correspondam aos mostrados na Tabela 4.

Z	C
0	1
0	0
1	1

Tabela 4: Casos de teste para a operação soma.

Referências

Charles Lin, 2003, Half Adders, Full Adders, Ripple Carry Adders, <http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Comb/adder.html>

Ting, Wu Shin & Fazanaro, Felipe Ieda, 2010, TUTORIAL Altera Quartus II e Kit de Desenvolvimento DE1, ftp://ftp.dca.fee.unicamp.br/pub/docs/ea773/Tutorial_QuartusII.pdf

ALTERA, 2006, Cyclone II FPGA Starter Development Board, https://www.altera.com/en_US/pdfs/literature/manual/mnl_cii_starter_board_rm.pdf