

Documentação EP1 - Álgebra Booleana

Gabriel de Russo e Carmo 9298041
Germano Hüning Neuenfeld 9298340
Luis Gustavo Bitencourt de Almeida 9298207
Victor Wichmann Raposo 9298020

Abril de 2016

1 Introdução

Este documento visa explicar como funcionam os circuitos de nossa Arithmetic Logic Unit (ALU), primeira parte do projeto de MAC0329 - Álgebra Booleana e Circuitos Digitais, ministrada pelo professor Junior Barrera.

Todo número a de n bits mencionado deste ponto em diante deve ser entendido como

$$a = a_{n-1}a_{n-2}\dots a_1a_0$$

onde a_{n-1} e a_0 são os bits mais e menos significativos, respectivamente. O bit *overflow* é 1 caso haja um overflow aritmético. Todos os números estão representados em complemento de dois.

2 Circuitos

2.1 main

É um exemplo de uma ALU em funcionamento completo.

2.2 ALU

Circuito principal do projeto. Recebe 2 números de 8 bits a e b , e um número de 3 bits s , que funciona como seletor da operação (soma, subtração, multiplicação, quociente ou resto), conforme especificado. A seleção é feita com um multiplexer do Logisim.

2.3 Half adder

Somador simples de dois bits a e b . Devolve a soma *sum* e o carry *cout*.

2.4 Full adder

Somador completo de bits que recebe além dos bits a e b também um bit carry *cin* e devolve a soma *sum* e o carry *cout*.

2.5 8-bit adder

Somador de dois números a e b de oito bits cada. Devolve a soma $a + b$ e um bit *overflow*.

2.6 2's complement

Recebe um número a de oito bits devolve seu oposto $-a$.

2.7 8-bit subtractor

Subtrai dois números a e b de oito bits cada, a partir da soma $a + (-b)$. Devolve a subtração $a - b$ e um bit *overflow*.

2.8 8-bit multiplier

Multiplica dois números a e b de oito bits cada, a partir de diversas somas e sucessivos left shifts (algoritmo padrão de multiplicação). Devolve o produto ab e um bit *overflow*.

2.9 8-bit divider

Divide dois números $|a|$ e $|b|$ de oito bits cada, a partir de diversas subtrações condicionais e sucessivos left shifts. Devolve o quociente a/b , o resto $a \% b$ (com sinais determinados pelo divisor e dividendo) e um bit *overflow*.

2.10 8-bit absolute

Recebe um número a de oito bits e devolve seu absoluto $|a|$.

2.11 8-bit iszero

Recebe um número a de oito bits e devolve 1 se $a = 0$ ou 0 caso contrário.

2.12 8-bit left shift

Recebe um número a de oito bits e um bit *over in*. Devolve $b = a_6a_5a_4a_3a_2a_1a_00$, ou seja, a deslocado para esquerda uma posição e um bit *overflow*. O bit *over in* serve para melhor organização do circuito de multiplicação.

2.13 8-bit can subtract

Recebe dois números a e b de oito bits cada. Devolve $a - b$ se $a - b \geq 0$ ou a caso contrário. Devolve também um bit que indica se a subtração foi efetuada (1 em caso positivo).

2.14 AND 8:1

Recebe um número a de oito bits e um bit b . Devolve o AND entre a e $bbbbbbb$.