

Projeto 02

Neste projeto vamos construir uma unidade lógica e aritmética (ULA) que é uma estrutura importante dentro dos processadores. A ULA proposta vai efetuar operações de adição e subtração em complemento de 2 e a operação de complementação.

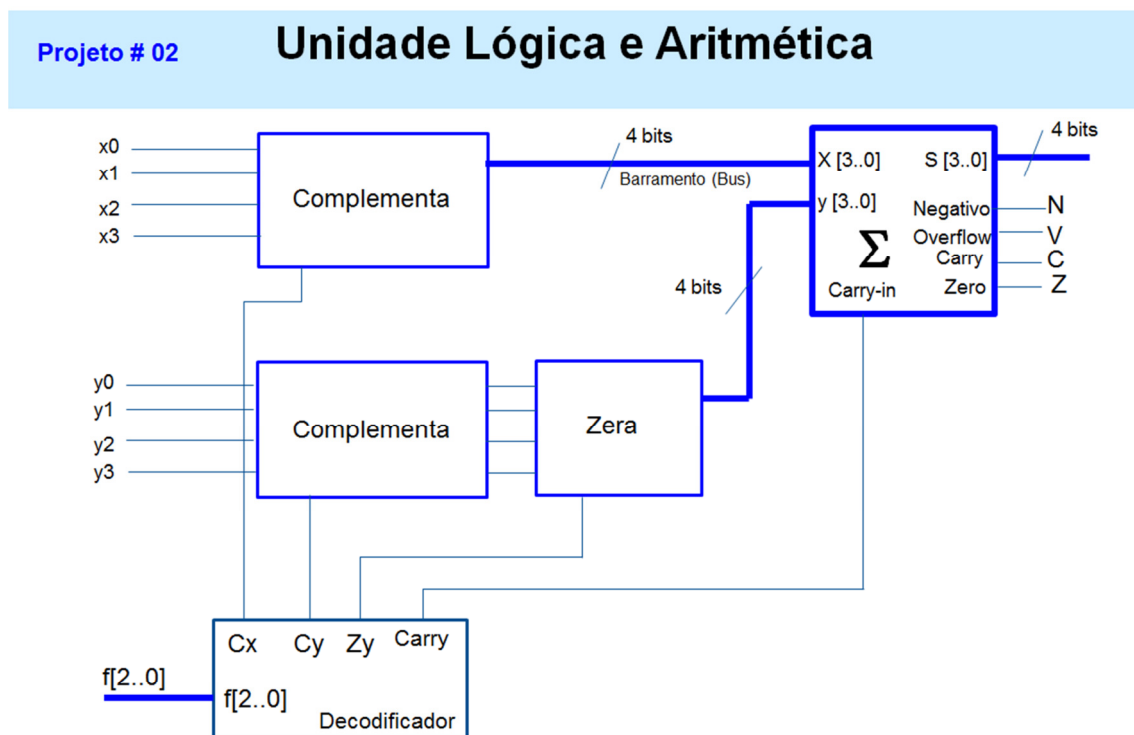
Os blocos têm as seguintes funções:

Complementa: Este bloco faz o complemento bit-a-bit da entrada (x0 a x3) e alimenta estes bits no barramento de 4 bits que está conectado na entrada da ULA (x[3..0], na notação do software Quartus II da Altera), de 4 bits.

Zera: se ativado pode zera ou não a entrada complementada que será a outra entrada (y[3..0]) de 4 bits da ULA. Estas entradas x e y nada mais são que os 4 bits de entrada de um somador de 4 bits.

Decodificador: o decodificador (que não é um decodificador clássico) é o circuito que vai ativar os blocos já mencionados e a ULA. A entrada é uma “função” de 3 bits f[2..0]. A combinação destes 3 bits vai dizer qual operação será efetuada na ULA. O circuito deve, em função da combinação dos 3 bits “setar” ou não as saídas Cx, Cy, Zy e Carry para possibilitar que os respectivos blocos sejam ativados. O circuito inteiro é um circuito combinacional, portanto, dada uma combinação de entrada x e y a saída da ULA de 4 bits será alterada, mas a operação a ser efetuada será controlada pelo decodificador. O carry que entra na ULA é usado em algumas operações aritméticas.

Se Cx for 1 significa que a entrada y será complementada pelo seu bloco complementa correspondente. Vamos falar da ULA e daí, voltamos, com exemplos.



ULA: Trata-se de um somador de 4 bits que executa a operação de soma de **x** e **y**. Se **y** for complementado antes, então será subtraído de **x**. Deste modo o resultado pode ser positivo ou negativo. Esta condição deve ser detectada por um circuito adicional da ULA para gerar o flag **N=1**, assim como as outras condições das flags, **V=1** se ocorrer overflow (como você sabe que houve overflow? Basta fazer um XOR do carry do bit mais significativo para fora com o carry do bit anterior para o mais significativo, como visto em aulas anteriores), **Z=1** se o resultado for nulo e **C=1** se o carry do mais significativo para fora ocorrer.

Então, se quisermos subtrair **y** de **x**. **F** deve conter uma combinação que faça o decodificador setar apenas **Cy** e então no bloco complementa (ligado ao **y**) a entrada será complementada. Assim, o complemento de **y** é somado com **x** e, portanto, a operação **x-y** será apresentada na saída e as flags correspondentes colocadas em 1. É preciso setar o Carry para fazer com que o complemento seja complemento de 2, na operação.

Vamos supor (o bit mais significativo é o **X₀**, mais à esquerda):

x = 0111

y = 0101

F[2..] = 010 (para indicar a operação desejada, de subtração). Neste caso há possibilidade de representação de $2^3 = 8$ operações.

Quando **x** e **y** forem alimentados na entrada e **F** for 010 **y** será complementado (obviamente, não será zerado) ficando $1010+1 = 1011$, **Cy** = 1 (seu decodificador vai setar **Cy** porque **F** = 010) e a saída será: $0111 + 1011 = 0010$ e ocorrerá um carry, portanto **C** = 1 e nenhuma outra flag será ativada (não é zero, não ocorreu overflow e não é negativo). Esta seria, então uma operação aritmética da ULA.

Suponha que você queira agora, apenas complementar o **x** e que o código de operação seria **F** = 111. Vamos supor **x** = 0011. A função **F** = 111 deve gerar o sinal **Zy** = 1 para zerar **y**, **Cx** = 1 para complementar **x**. O resultado é que a ULA irá somar o complemento de **x** com zero e, portanto, o resultado será o complemento de **x** na saída.

x = 0011, **Cx** = 1, então na entrada correspondente da ULA ficaria 1101. Qualquer que seja **y**, na entrada correspondente da ULA ficaria 0000. O resultado seria então 1101 na saída que é o complemento de 2 de 0011. Sua ULA acaba de fazer uma operação lógica.

Você sabe fazer qualquer um destes circuitos combinacionais. Todos são desprovidos de memória e, portanto, dado o valor de **F** e das entradas a saída será computada.

Faça outros exemplos para gerar todas as opções dentro da ULA e flags.

Boa sorte!!!