

Técnicas Digitais

Prof. José Luís Güntzel guntzel@ufpel.edu.br – semestre 2003/1

Projeto de uma Unidade Lógico-Aritmética (ULA)

Uma unidade lógico-aritmética (ULA) é responsável pelas operações lógicas e aritméticas básicas num processador. As operações aritméticas tipicamente realizadas por uma ULA são adição, subtração, incremento e decremento. Dentre as operações lógicas citam-se o E, o OU, identidade e complemento. Note que as primeiras duas operações lógicas são definidas entre os pares de bits, cada um pertencendo a um dos operandos. Já as duas últimas operações lógicas são definidas para os bits de (somente) um dos operandos.

Como todas as operações aritméticas se baseiam na adição, pode-se projetar uma ULA acrescentando-se alguns circuitos lógicos especiais às entradas de um somador (*ripple-carry* ou *carry lookahead*). Os circuitos lógicos usados na realização das operações lógicas às vezes recebem o nome de **extensores lógicos (EL)** enquanto que aqueles usados na realização das operações aritméticas às vezes são chamados **extensores aritméticos (EA)**. A figura 1 mostra um somador para dois números de 4 bits (A e B) com os extensores lógicos e aritméticos acoplados às suas entradas. Note que cada extensor recebe como entrada somente um dos bits de cada número (A e B).

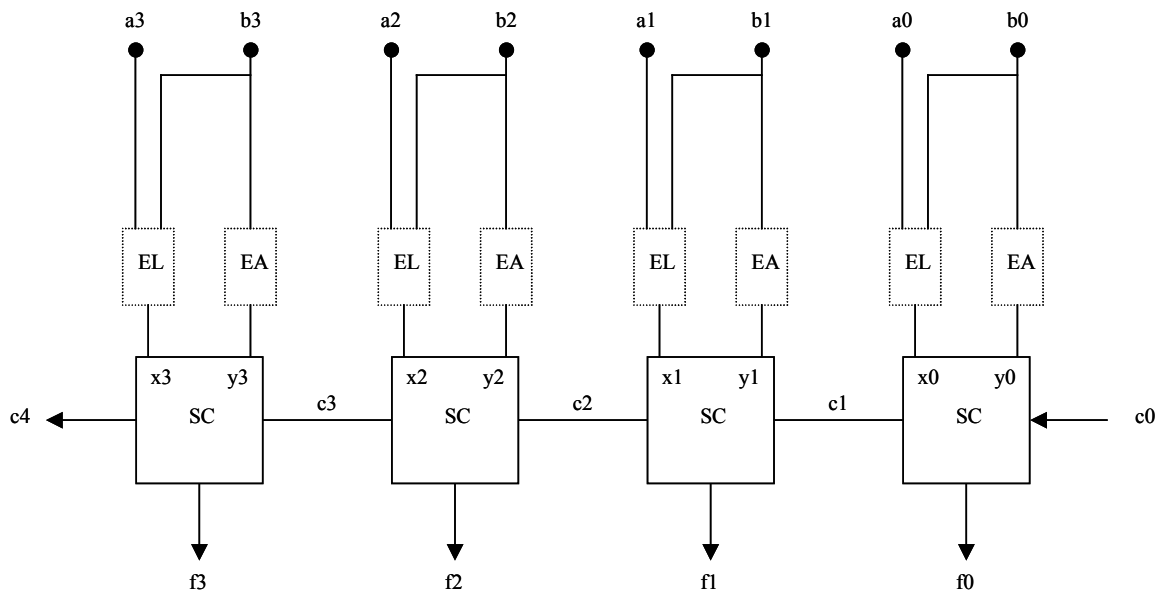


Figura 1 - Somador para dois números de 4 bits com extensores lógicos (EL) e aritméticos(EA).

Suponha que se deseje projetar uma ULA capaz de realizar 4 operações lógicas e 4 operação aritméticas. Assim, estaremos usando os seguintes sinais de controle: M, S0 e S1. Se M=0, as variáveis S1 e S0 selecionam uma dentre as 4 operações lógicas disponíveis. Se M=1, S1 e S0 selecionam uma dentre as 4 operações aritméticas. A tabela 1 mostra as operações desejadas bem como os assinalamentos das variáveis M, S1 e S0 que selecionam cada uma das 8 operações disponíveis.

Tabela 1 - Operações desejadas para a ULA a ser projetada.

| M | S1 | S0 | nome da função | F | X | Y | C0 |
|---|----|----|----------------|--------|--------|----------|----|
| 0 | 0 | 0 | complementa | A' | A' | 0 | 0 |
| 0 | 0 | 1 | E | A E B | A E B | 0 | 0 |
| 0 | 1 | 0 | identidade | A | A | 0 | 0 |
| 0 | 1 | 1 | OU | A OU B | A OU B | 0 | 0 |
| 1 | 0 | 0 | decrementa | A-1 | A | todos 1s | 0 |
| 1 | 0 | 1 | soma | A+B | A | B | 0 |
| 1 | 1 | 0 | subtrai | A+B'+1 | A | B' | 1 |
| 1 | 1 | 1 | incrementa | A+1 | A | todos 0s | 1 |

O ponto de partida para o projeto da ULA em questão será o somador de 4 bits, cujas entradas estaremos denominando X (x_3, x_2, x_1, x_0) e Y (y_3, y_2, y_1, y_0), além da entrada c_0 . A saída do somador será denominada F (f_3, f_2, f_1, f_0) e coincide com a saída da ULA. Por outro lado, a ULA a ser projetada recebe como entrada dois números de 4 bits, os quais estaremos denominando A (a_3, a_2, a_1, a_0) e B (b_3, b_2, b_1, b_0).

O projeto em questão será dividido em duas partes: projeto dos **extensores aritméticos** e projeto dos **extensores lógicos**. Os extensores aritméticos devem realizar as operações aritméticas, sendo utilizados ao se fazer $M=1$. Examinando a tabela 1 nota-se que para as situações em que $M=1$ a entrada X do somador deve receber o valor A (sempre), enquanto que a entrada Y recebe 1s, B, B' ou 0s, conforme for a combinação de S1 e S0. Portanto, cada extensor aritmético irá operar sobre um bit de B, de modo a gerar o resultado conforme mostrado na coluna Y da tabela 1. Além disso, deve-se notar que quando a operação solicitada corresponder a uma operação lógica ($M=0$), Y deve valer 0 (ou seja, todos os bits de Y devem ser iguais a zero). A partir deste detalhamento, pode-se montar a tabela-verdade para o circuito de um extensor aritmético (tabela 2), da qual deriva-se o mapa de Karnaugh (tabela 3) que nos conduz à equação simplificada.

Tabela 2 – Tabela-verdade para o extensor aritmético.

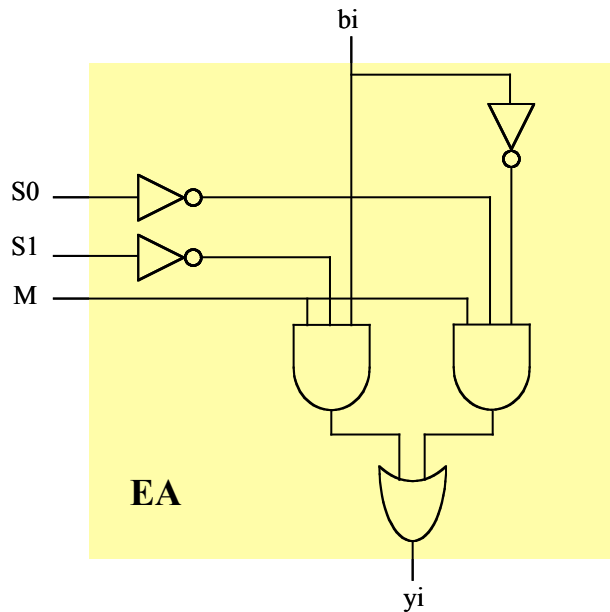
| S1 | S0 | b_i | y_i |
|----|----|-------|-------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Tabela 3 – Mapa de Karnaugh para o extensor aritmético.

| S1S0 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| b_i | | | | |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 |

A equação minimizada, obtida a partir do mapa de Karnaugh será $y_i = S1'.b_i + S0'.b_i'$. Note que, para efeitos de simplificação, omitiu-se a variável de entrada M. No entanto, tal variável deve estar presente. Para tanto, observamos na tabela 1 que sempre que $M=1$ a operação selecionada é do tipo aritmética. Por outro lado, se $M=0$, $Y=0$, ou seja, cada bit de Y deve valer zero. Assim, deve-se

fazer o E entre M e cada produto da equação anterior, o que conduz a seguinte equação: $y_i = M.S1'.bi + M.S0'.bi'$. A figura que segue mostra o circuito lógico referente a essa equação, o que



corresponde ao extensor aritmético.

Figura 2 – Circuito de um extensor aritmético (EA).

Passemos agora ao projeto dos **extensores lógicos**. Analisando-se a tabela 1, nota-se que $Y=0$ e $c0=0$ quando $M=0$. Estas condições são consequência do fato de se ter definido que todas as operações lógicas são executadas exclusivamente pelos extensores lógicos, de modo que o somador não é utilizado quando $M=0$. Porém, como X (que corresponde às saídas dos extensores lógicos) entra no somador, será necessário fazer $Y=0$ e $c0=0$ de modo a permitir que X passe pelo somador sem ser alterado ($F=X$). Deve-se notar ainda que quando $M=1$, $X=A$. As condições descritas neste parágrafo podem ser transcritas para a tabela-verdade que segue, a qual permite determinar uma expressão lógica para X em função de M , $S1$, $S0$, A e B .

Tabela 4 – Tabela-verdade para o extensor lógico.

| M | S1 | S0 | x_i |
|---|----|----|----------|
| 0 | 0 | 0 | a_i' |
| 0 | 0 | 1 | $a_i.bi$ |
| 0 | 1 | 0 | a_i |
| 0 | 1 | 1 | a_i+bi |
| 1 | ? | ? | a_i |

Tabela 5 – Mapa de Karnaugh para o extensor aritmético.

| | | M=0 | | | | M=1 | | | |
|------|-----------|-----|----|----|----|-----|----|----|----|
| | | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
| S1S0 | $a_i b_i$ | | | | | | | | |
| 00 | | 1 | | | | | | | |
| 01 | | 1 | | 1 | | | | | |
| 11 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | | | | 1 | 1 | 1 | 1 | 1 | 1 |

A partir do mapa de Karnaugh mostrado pela tabela 5 chega-se a seguinte expressão minimizada para x_i : $x_i = M'.S1'.S0'.ai' + M'.S1.S0.bi + S0.ai.bi + S1.ai + M.ai$. O circuito lógico correspondente pode ser visto na figura 3.

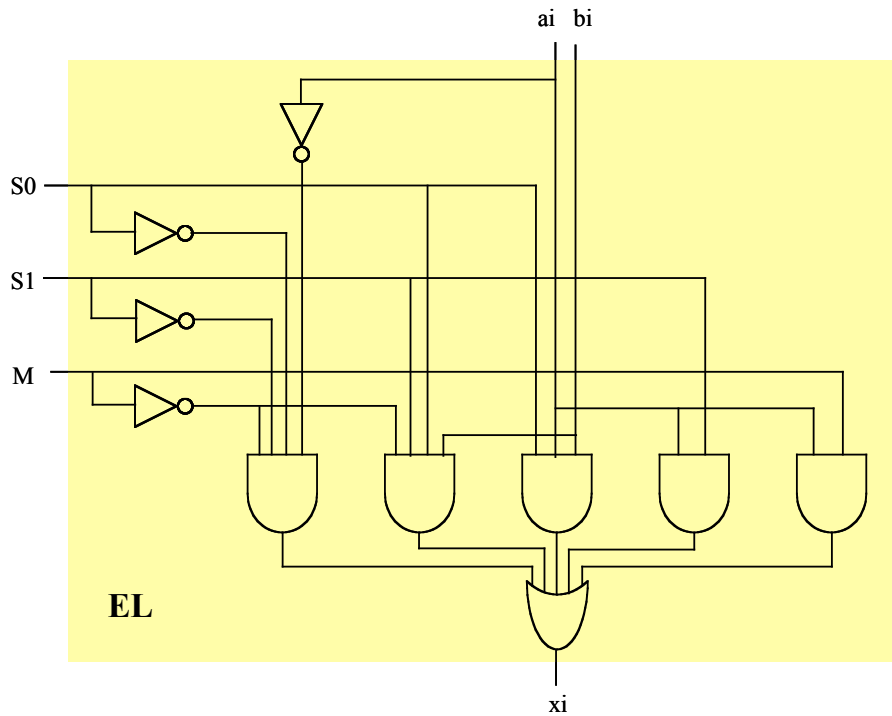


Figura 3 – Circuito de um extensor lógico (EL).

Feito o projeto dos extensores lógicos e aritméticos, deve-se encontrar a equação lógica que descreve o comportamento da variável c_0 . Pode-se derivar a equação de c_0 observando-se a tabela 1. Esta equação é $c_0 = M.S1$, ou seja, o E entre M e S1. Para finalizar o projeto da ULA, observa-se que, como partimos de um somador que pode ser configurado para realizar subtrações, o sinal de overflow será formado da mesma maneira que em um somador/subtrator, ou seja, mediante a colocação de uma porta OU-exclusivo (XOR) entre o último e o penúltimo transportes, conforme mostra a figura 4.

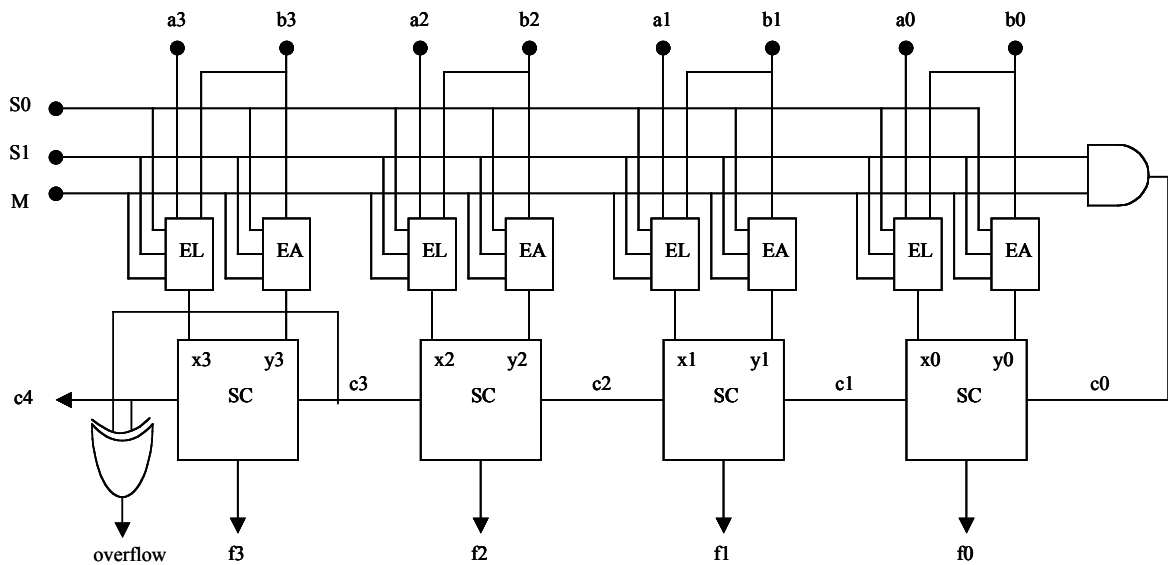


Figura 4 – Diagrama da ULA projetada (com 4 bits).

Caso se deseje uma ULA para operar sobre números com mais do que 4 bits, basta que se conecte o número desejado de somadores completos e extensores lógicos e aritméticos, sempre seguindo o esquema mostrado na figura 4. A figura 5 mostra um símbolo possível para a ULA projetada. Neste símbolo, o número de bits foi omitido, de modo a torna-lo genérico.

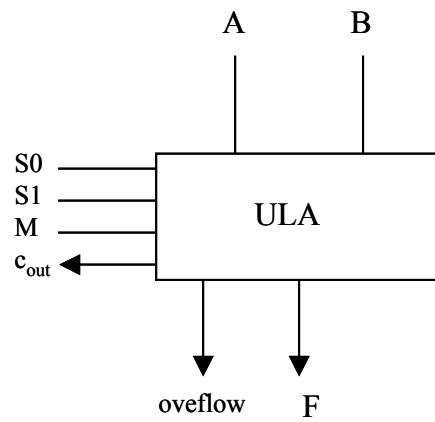


Figura 5 – Um possível símbolo para a ULA projetada.