

Projeto de uma Unidade Lógica - Aritmética de 8 bits

Denise Carvalho de Freitas¹, Fernanda Costa de Sousa¹

¹Universidade Federal do Ceará (UFC)

Russas – CE – Brasil

denisecarvalhofrts@gmail.com, fernanda.costa@outlook.com

Abstract. *This work presents the Project and development of an Arithmetic Logic Unit (ALU) executed in the CMOS technology 0.35 μ m process. This ALU accomplishes arithmetic sum, subtraction, comparison of operands and logic functions AND and OR operations for a simple architecture. Our general objective is the academic development in the computer Architecture discipline.*

Resumo. *Este trabalho apresenta o projeto e desenvolvimento de uma Unidade Lógica e Aritmética (ULA) implementada no processo de 0.35 μ m da tecnologia CMOS. Esta ULA realiza operações aritméticas de soma, subtração, comparação (maior, menor e igual) entre operandos e funções lógicas AND e OR para uma arquitetura simples. Nosso objetivo geral é o desenvolvimento acadêmico na disciplina de Arquitetura de computadores.*

1. Introdução

A unidade lógica e aritmética (ULA) ou ALU (Arithmetic Logic Unit) é responsável pelas operações lógicas e aritméticas em um processador. Este trabalho tem como objetivo geral apresentar o estudo e desenvolvimento de uma ULA com base em estudo bibliográfico para fins acadêmicos.

A ULA foi proposta por John Von Neumann em 1946. Deste então o seu desenvolvimento tem sido de vital importância para o aceleração do processamento de dados e operações matemáticas mais rapidamente.

Atualmente uma ULA pode realizar diversas operações, mas neste trabalho apresentaremos apenas as operações aritméticas de soma e subtração, comparação (maior, menor e igual) entre operandos e as operações lógicas AND e OR.

Todos os circuitos da ULA foram construídos e simulados com o simulador lógico Logisim onde foram realizados testes com todos os circuitos.

2. Visão geral de uma ULA

A unidade lógica e aritmética (ULA) é um circuito digital que realiza operações lógicas e aritméticas, é também uma peça fundamental da unidade central de processamento (CPU) e de microprocessadores. É conhecida como uma grande calculadora eletrônica e quando os primeiros computadores modernos foram construídos o conceito de ULA já estava disponível.

A tecnologia da ULA foi primeiramente utilizada em relés, logo depois em válvulas, depois em circuitos integrados, seus circuitos passaram a ser implementados com a tecnologia de semicondutores.

3. ULA de 8bits e suas funções

Segundo GÜNZEL, 2003, toda operação aritmética realizada na ULA é baseada na operação de adição.

A proposta dessa ULA é apresentada da seguinte forma, temos uma unidade aritmética constituída de oito blocos somadores completos, interligados pelo sinal de vai-um (carry para soma e borrow para subtração), sendo capaz de realizar operações de soma e subtração com números binários de até oito bits.

A unidade lógica é composta pela combinação de 8 blocos lógicos que realizam as operações AND e OR com palavras de até 8 bits de forma paralela.

O modelo de ULA estudado possui duas entradas de 8 bits para cada um dos dois operandos, uma saída de 8 bits para o resultado, entradas de sinais controle (seleção) que sinalizam para a ULA qual operação deve ser realizada e também sinais de saída que indicam o resultado da operação.

Ao ajustar os sinais de controle (seleção), o circuito interno da ULA é preparado para realizar a operação entre os operandos dispostos nas portas de entrada, o resultado estará disposto na porta de saída e o estado da operação nas saídas de sinalização de estado.

Além do objetivo geral, este trabalho tem como finalidade específica trabalhar com operandos e resultados de 8 bits, tendo como especificações as seguintes características:

- As entradas são A e B, e a saída de resultados S, todas sendo de 8 bits.
- Operações na forma: $A \text{ operação } B = S$.
- Operações aritméticas: Soma e Subtração.
- Operações de comparação: Maior, Menor e Igual.
- Operações lógicas: lógica AND (E) e OR (OU).

Seleção de função: Palavra de 3 bits utilizada para configurar a operação na ULA, conforme tabela.

Tabela 1- Palavras de seleção de operação na ULA.

Código (C2, C1, C0)	Operação
000	Soma
001	Subtração
010	Comparação Igual
011	Lógica OR
100	Lógica AND
101	Comparação Maior
110	Comparação Menor

3.1. Operação soma

A base para o projeto da unidade de soma e subtração (USS) é módulo somador, que foi adaptado do somador de um bit proposto na literatura por Fregni e Saraiva (1995) e Tocci, Widmer e Moss (2007). O Somador de um bit realiza adição em uma única coluna binária da cadeia de bits como mostra a figura 3.

A operação de adição inicia ao somar-se os bits menos significativos (A0 ou B0) da primeira e da segunda palavra, sendo assim $1+1=10$, ou seja, a soma dos dois bits menos significativos resulta um bit menos significativo igual a 0 e o bit vai-um (carry-out) igual a 1.

O bit vai-um da coluna anterior é recebido na próxima coluna como vem-um que chega (carry-in) e é somado aos bits da segunda coluna, dessa forma $1+1+0=10$, ou seja, a soma de C_i (carry-in) com o dos bits resulta em zero e um bit C_o (carry-out) igual a um, e assim até a soma dos bits mais significativos.

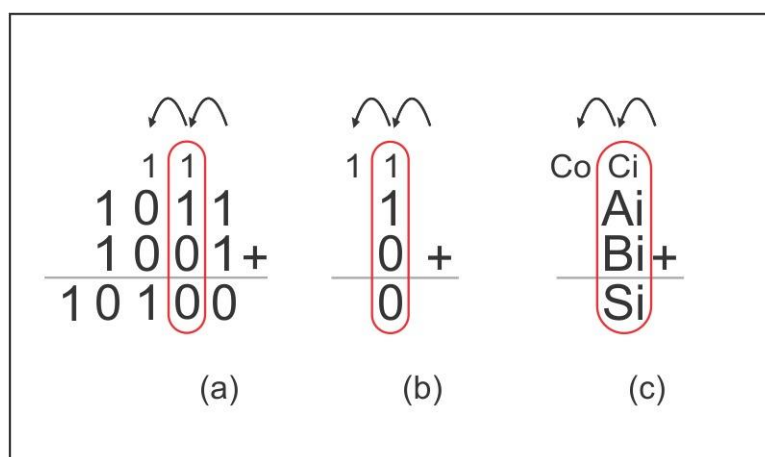


Figura 1. Estudo e modelo do somador bit-a-bit proposto por Fregni e Saraiva (1995) e Tocci, Widmer e Moss (2007), onde (a) representa a soma paralela, (b) uma única coluna da soma com detalhe do bit recebido (carry-in) da soma anterior e o bit que vai para a soma seguinte (carry-out) e (c) modelo algébrico.

A partir da figura 1 podemos observar que as colunas da soma binária, cada uma representando um módulo somador são inter-relacionadas pelo bit vai-um (carry-out) que se propaga entre elas, onde o bit vai-um que tem origem na coluna anterior é o C_i (carry-in) é somado com os operandos A e B da posição i. Já o bit vai-um que tem como destino a próxima coluna é o C_o (carry-out).

As equações para S_i e C_o são vistas nas equações 1 e 2:

$$S_i = (A_i \oplus B_i) \oplus C_i \quad (1)$$

$$C_o = A_i B_i + C_i (A_i \oplus B_i) \quad (2)$$

Para a operação de soma, foi criado um circuito de soma com três entradas sendo elas A, B e Carry-in e duas portas de saída soma e Carry-out, esse circuito será encapsulado no circuito soma-8bits para somar os 16bits de entrada (8bits de entradas A e 8bits de entradas B), nesse circuito os bits de entrada são interligados no encapsulamento, a soma será realizada do bit menos significativo (A0 e B0) para o mais significativo (A7 e B7), o bit que representa carry-out estará ligado com o próximo carry-in, esse circuito armazenará o resultado (saídas de S0 à S7) mais o overflow (carry-out do último encapsulamento). O carry-in do primeiro encapsulamento está ligado a um fio terra, pois como é a primeira operação, não possui o bit vai-um.

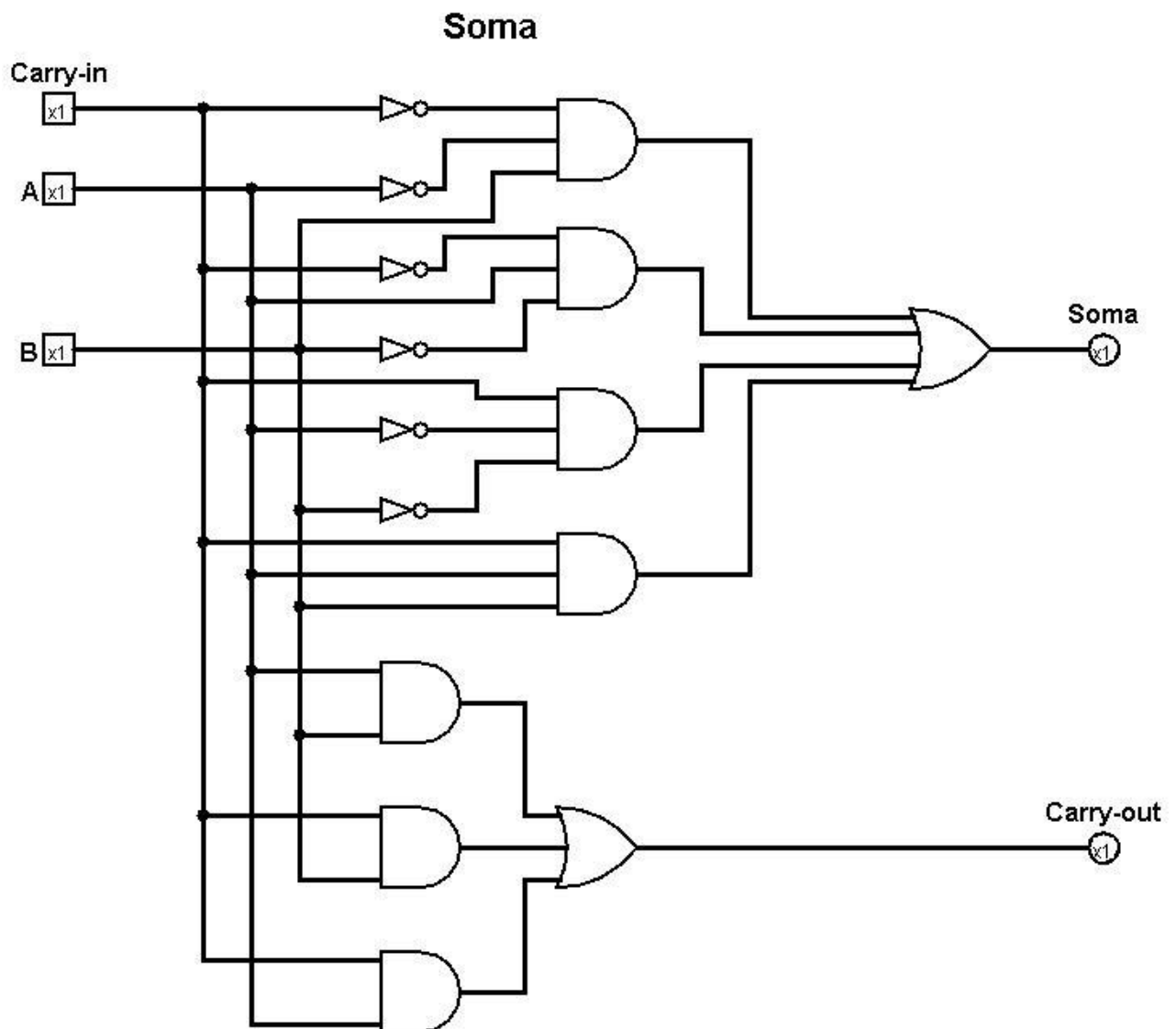


Figura 2. Circuito Soma.

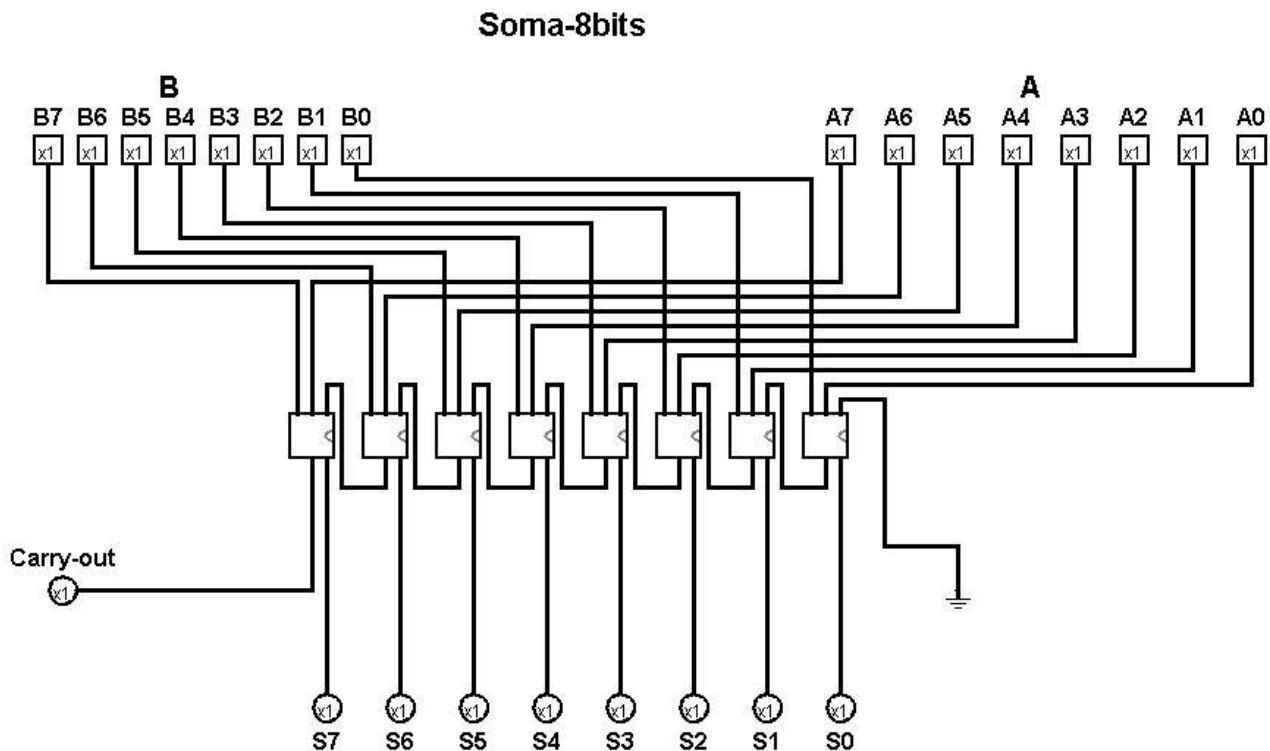


Figura 3. Circuito Soma-8bits.

3.2. Operação subtração

A operação subtração consiste em realizar a subtração entre os bits menos significativos da primeira palavra (A) e da segunda (B), exemplo $1-1=11$, então obtém-se um resultado no bit menos significativo igual a 1 e bit pede um (borrow-out) igual a 1.

O bit pede um da coluna anterior é recebido na próxima coluna como vai-um que chega (borrow-in) e é subtraído aos bits da segunda coluna, dessa forma $1-1-1=11$, ou seja, a subtração de Bi (borrow-in) com o dos bits resulta em um e um bit Bo (borrow-out) igual a um, e assim até a subtração dos bits mais significativos.

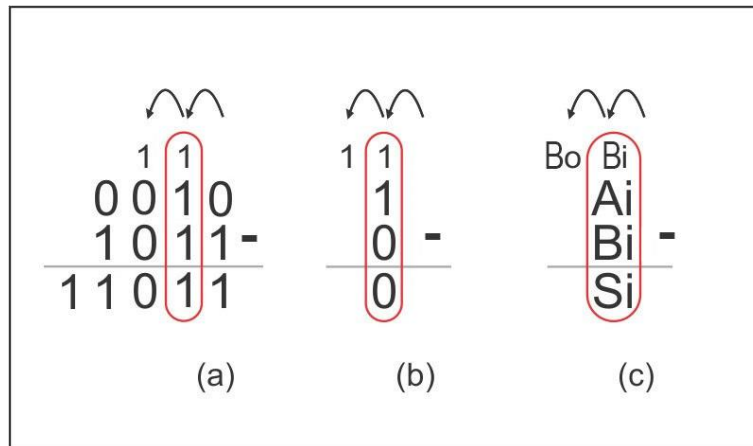


Figura 4. Representação da subtração paralela, (b) uma única coluna da subtração com detalhe do bit recebido (borrow-in) da subtração anterior e o bit que vai para a subtração seguinte (borrow-out) e (c) modelo algébrico.

A partir da figura 4 podemos observar que as colunas da subtração binária, cada uma representando um módulo substrator são inter-relacionadas pelo bit pede um (borrow-out) que se propaga entre elas, onde o bit pede um que tem origem na coluna anterior é o Bi é subtraído com os operandos A e B da posição i. Já o bit pede um que tem como destino a próxima coluna é o Bo.

As equações para Si e Bo são vistas nas equações 1 e 2:

$$B_o = Borrow_o$$

$$B_i = Borrow_i$$

$$S_i = (A_i - B_i) - Borrow_{i(1)}$$

$$Borrow_o = A_i B_i - Borrow_i (A_i - B_i)(2)$$

Para realizar a operação de subtração, foram necessários dois circuitos o subtração que realiza a operação entre 3 bits de entrada um desses bits é o borrow-in e os outros dois são A e B, como resultado temos as saídas subtração e borrow-out, esse circuito foi encapsulado assim como o de subtração para ser possível realizar uma operação de subtração com duas entradas de 8bits cada, e o segundo circuito necessário é Subtração-8bits que é onde vai ser utilizado encapsulamentos do circuito subtração, esse circuito calcula a subtração entre os bits das entradas A e B e armazena cada um dos resultados nas saídas S, esse circuito liga a porta de saída borrow-out com a próxima porta de entrada borrow-in. A última porta borrow-out significa o overflow na operação e a primeira porta borrow-in fica ligada a uma constante 0, pois não possui nenhum borrow-out antecedente à ela.

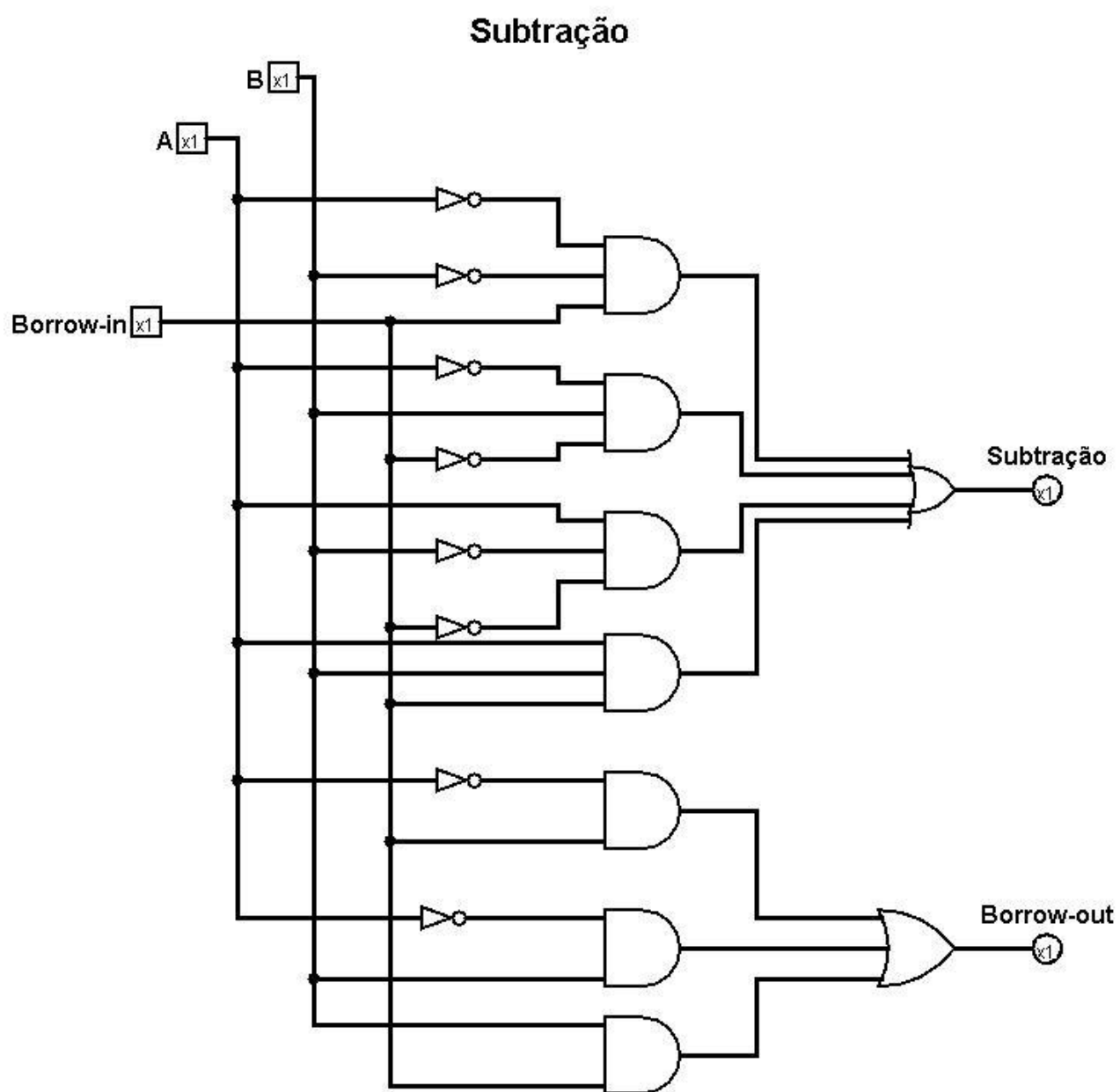


Figura 5. Circuito Subtração.

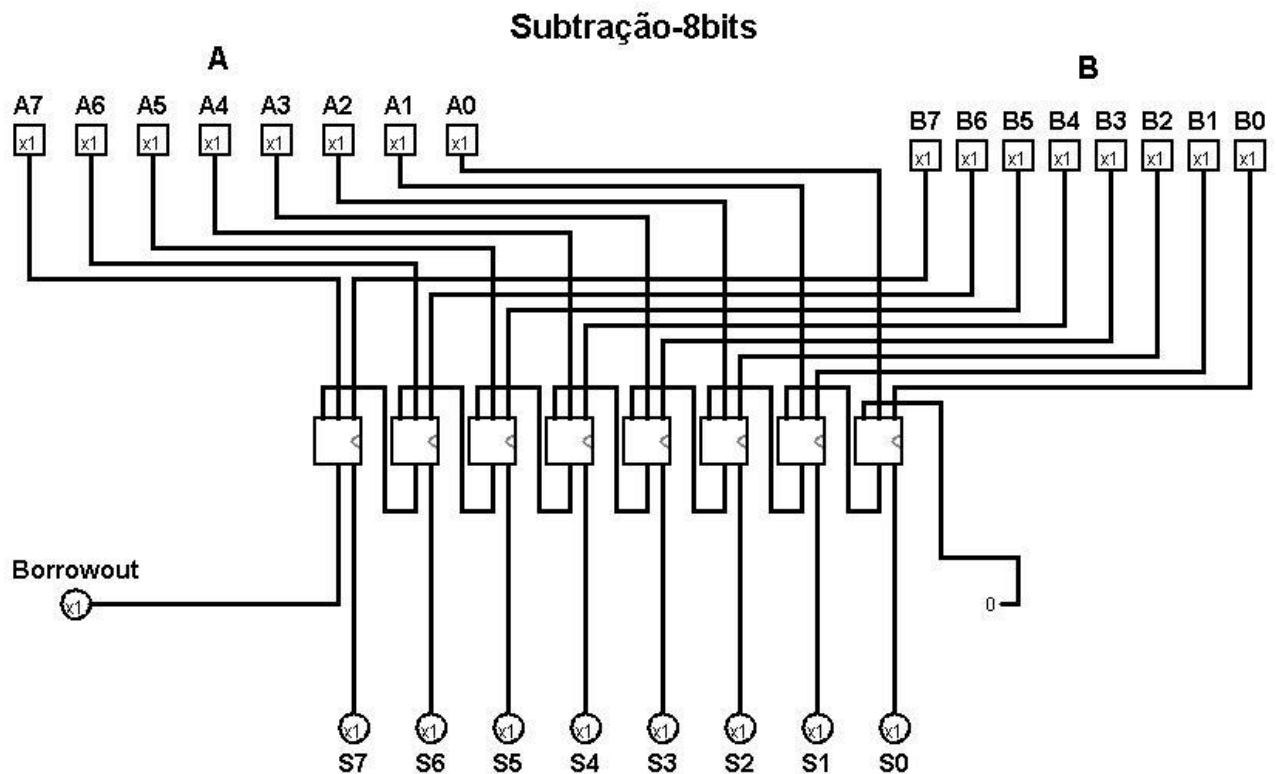


Figura 6. Circuito Subtração-8bits.

3.3. Operação de comparação: Igual

Para a operação de comparação igual foi criado um circuito com duas entradas A e B, e uma porta de saída igual. Os sinais de entrada são interceptados e comparados, se forem iguais a saída receberá bit 1. Esse circuito foi encapsulado no circuito Igual-8bit que implementa a realização da comparação de igualdade dos bits de A0 a A7 e B0 a B7.

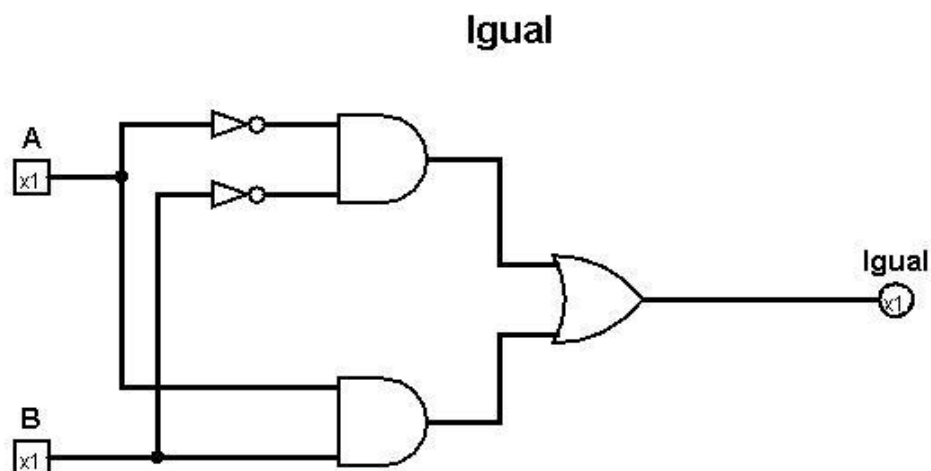


Figura 7. Circuito de comparação igual.

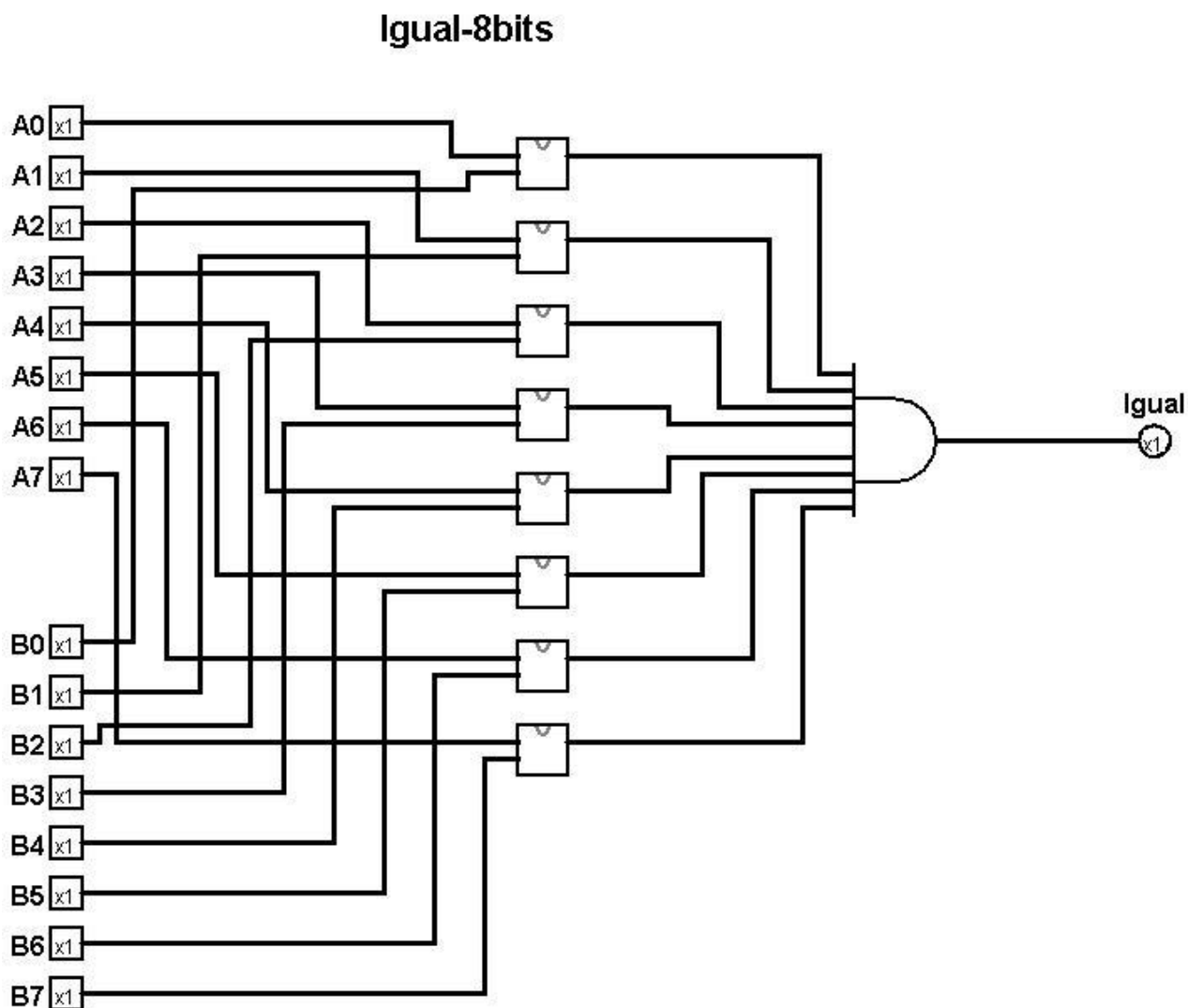


Figura 8. Circuito Igual-8 bit.

3.4. Operação de comparação: Menor

Para a operação de comparação Menor fizemos um circuito usando as portas NOT, AND e OR, que através desse circuito combinatório permite a comparação entre os bits de A e B. Neste circuito existem as entradas A e B, ambas com oito bits e uma saída que indica se A é menor que B. Se A for menor que B a saída terá resultado S igual a 1, caso contrário o resultado de S será 0.

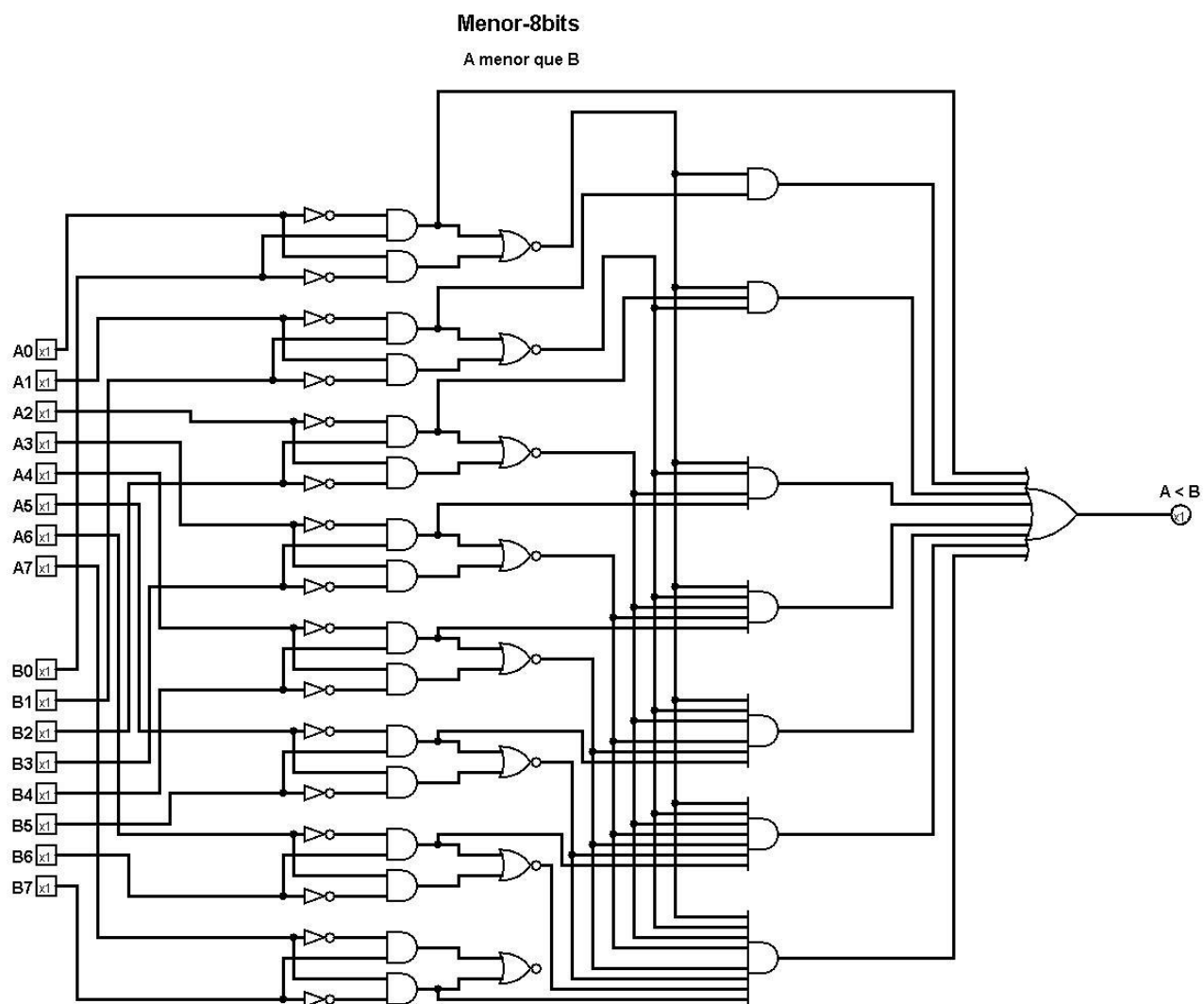


Figura 9. Menor-8 bit.

3.5. Operação de comparação: Maior

Da mesma forma que o circuito de comparação menor, neste circuito existem as entradas A e B, ambas com oito bits, e uma saída que neste circuito indica se A é maior que B. Se A for maior que B a saída terá resultado S igual a 1, caso contrário o resultado de S será 0.

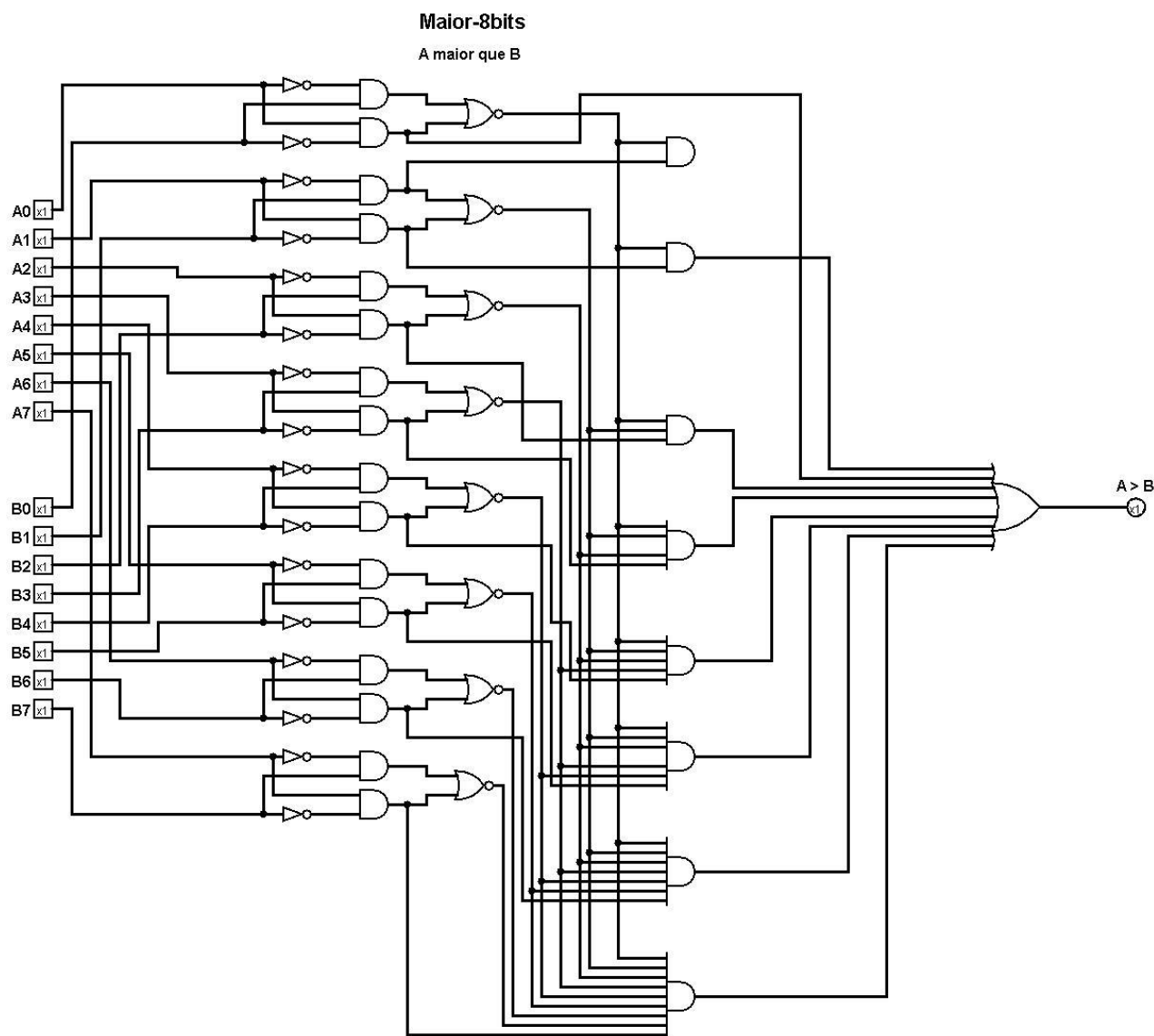


Figura 10. Maior-8 bit.

3.6. Modulo comparador

O modulo comparador é um circuito lógico de combinação que realiza a comparação de duas palavras binárias e gera uma saída (S) codificada que indica qual destas palavras (A e B) tem maior magnitude (TOCCI, WIDMER & MOSS, 2007).

O MC, opera em conjunto com a USS configurada para realizar operações de subtração. Ao ser realizada a operação de subtração entre os operandos A (A7 A6 A5 A4 A3 A2 A1 A0) e B (B7 B6 B5 B4 B3 B2 B1 B0) o MC monitora o resultado da operação e informa se A=B, se A<B ou se A>B.

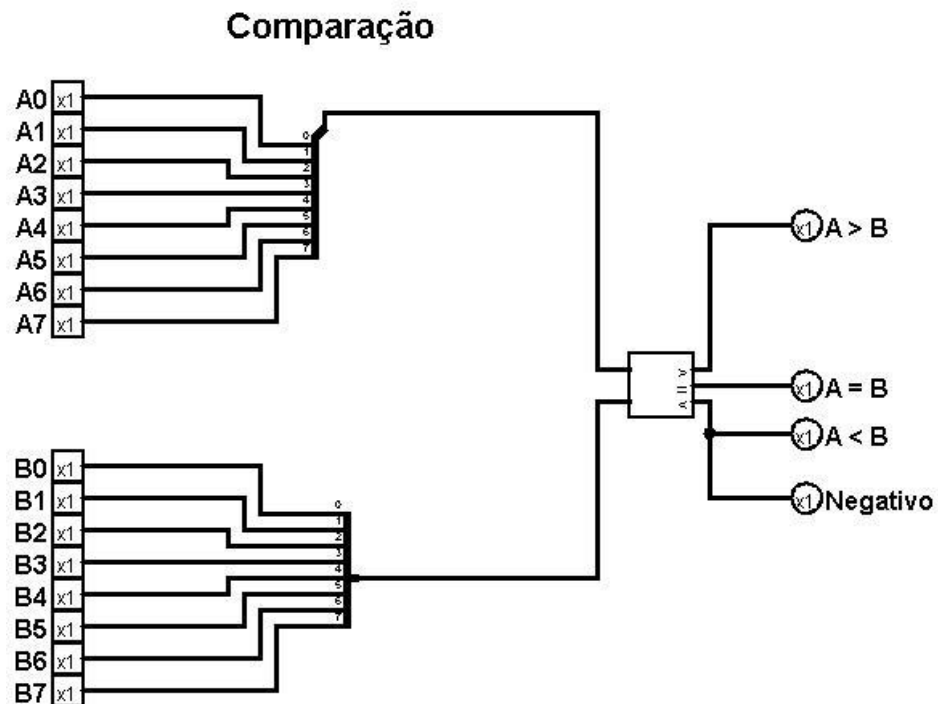


Figura 11. Circuito do Módulo comparador.

3.7. Unidade logica

As operações lógicas na ULA são realizadas em cadeias de bits, logo, quando se realiza uma operação lógica entre os operandos A e B de uma ULA, a operação é realizada bit-a-bit, entre todos os bits dos dois operandos.

A unidade lógica proposta realiza funções lógicas sobre os operandos A e B, conforme a configuração do sinal de controle (C), ou seja, o sinal de controle determina a função lógica que será realizada.

É ilustrado na tabela. 3 os sinais de controle da UL a partir do sinal de controle na porta C, onde foi configurada a unidade para realizar a função lógica *OR* ou *AND*.

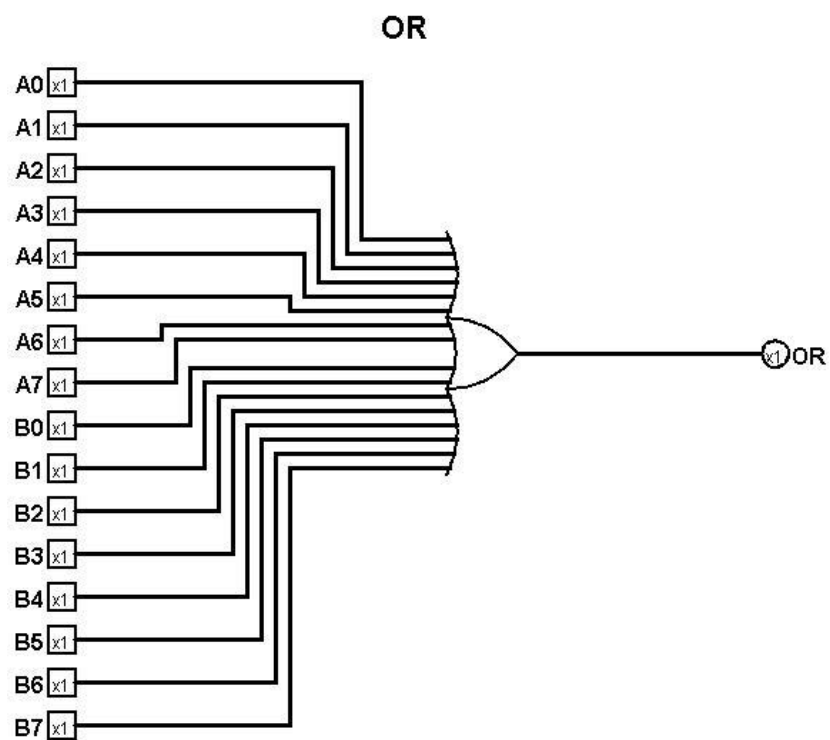


Figura 12. Circuito OR.

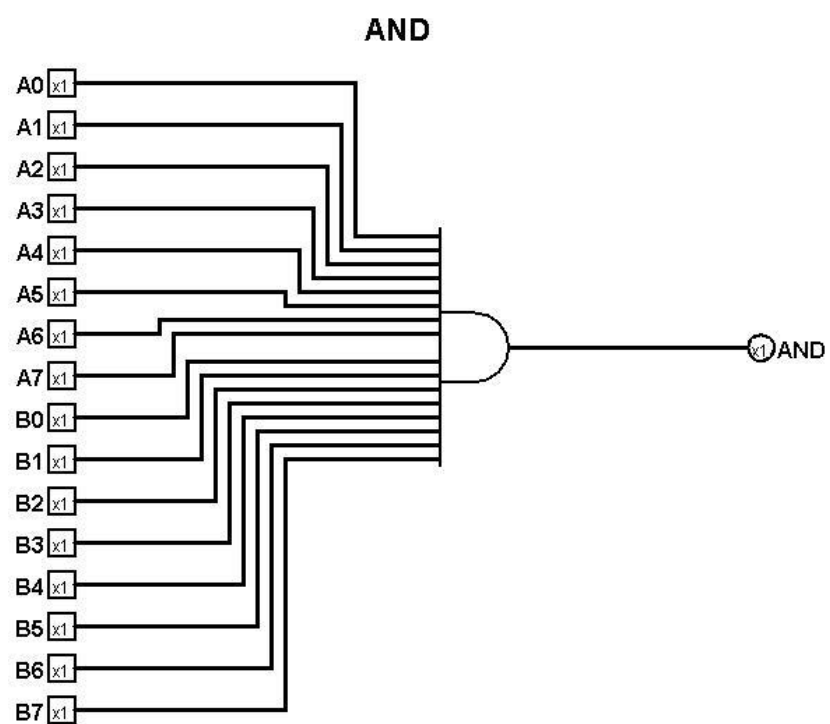


Figura 13. Circuito AND.

4. Topologia

As operações da ULA são divididas em circuitos modulares, sendo: soma, subtração, igual, maior, menor, AND e OR, essas operações estão ligadas a um multiplexador que possui um pino de 3bits para a seleção da operação e também oito saídas cada qual de 1bit que constitui o resultado da operação selecionada o resultado possui 8bits. Esse circuito possui também um extensor para as operações igual, maior, menor, AND e OR, para transformar 1bit de saída para 8bits.

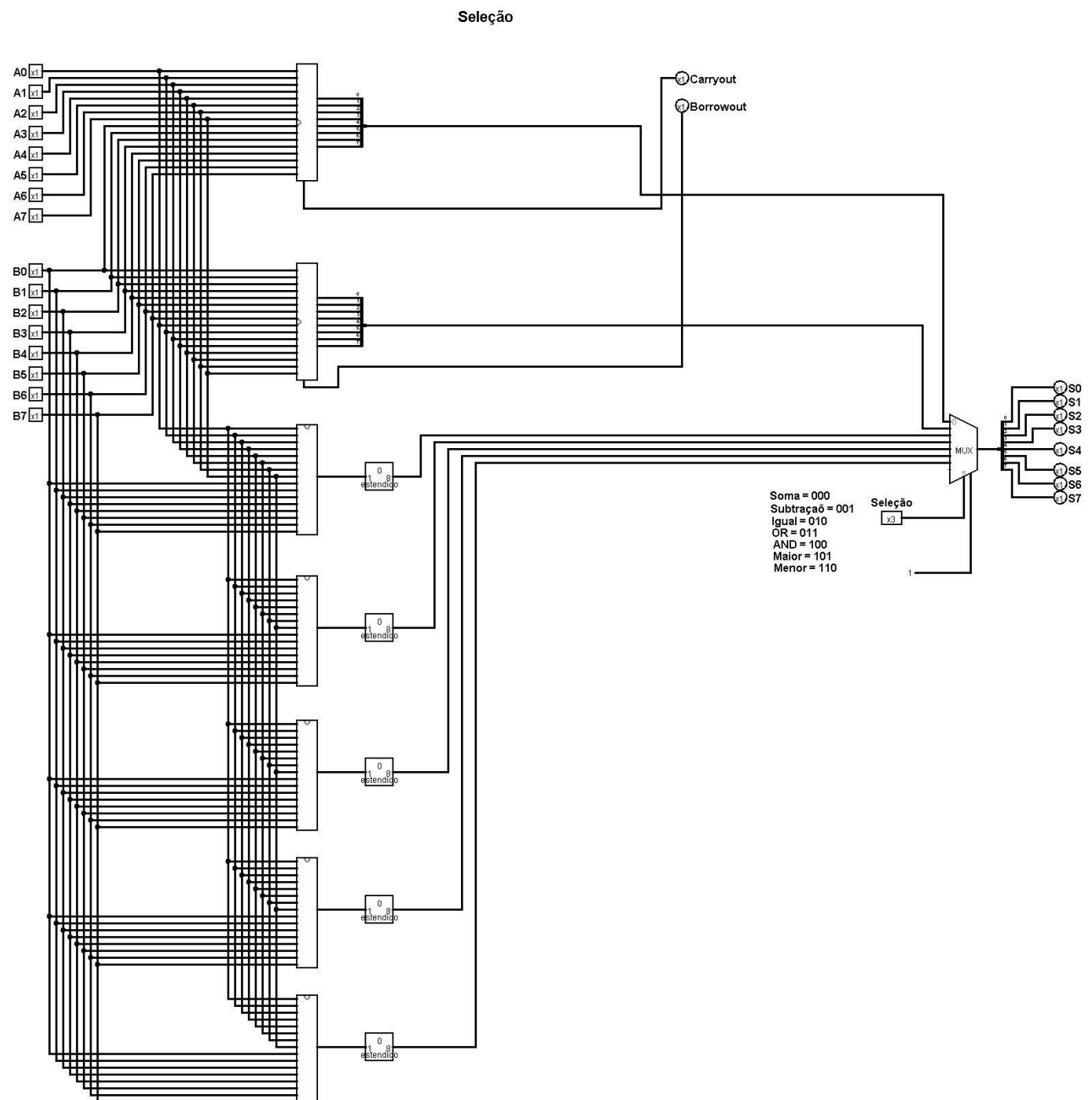


Figura 14. Circuito Seleção, possui um encapsulamento para cada operação.

Na ULA proposta também foi criado o circuito de operações onde foi encapsulado o circuito seleção onde ira sair as respostas da operação escolhida e também dois sinalizados sendo eles o carry-out e o borrow-out para se a operação soma ou subtração for escolhida sinalizar se na operação houve overflow, mas também irá encapsular o circuito de comparação onde haverá quatro sinalizadores que irá indicar se as entradas A e B são iguais, se A é maior que B, se A é menor que B e se o número é negativo.

O sinalizador da negação só fará sentido se a operação que estiver sendo realizada seja a subtração, pois a porta B será maior que a porta A e ao subtrair essas duas entradas o resultado será um valor negativo.

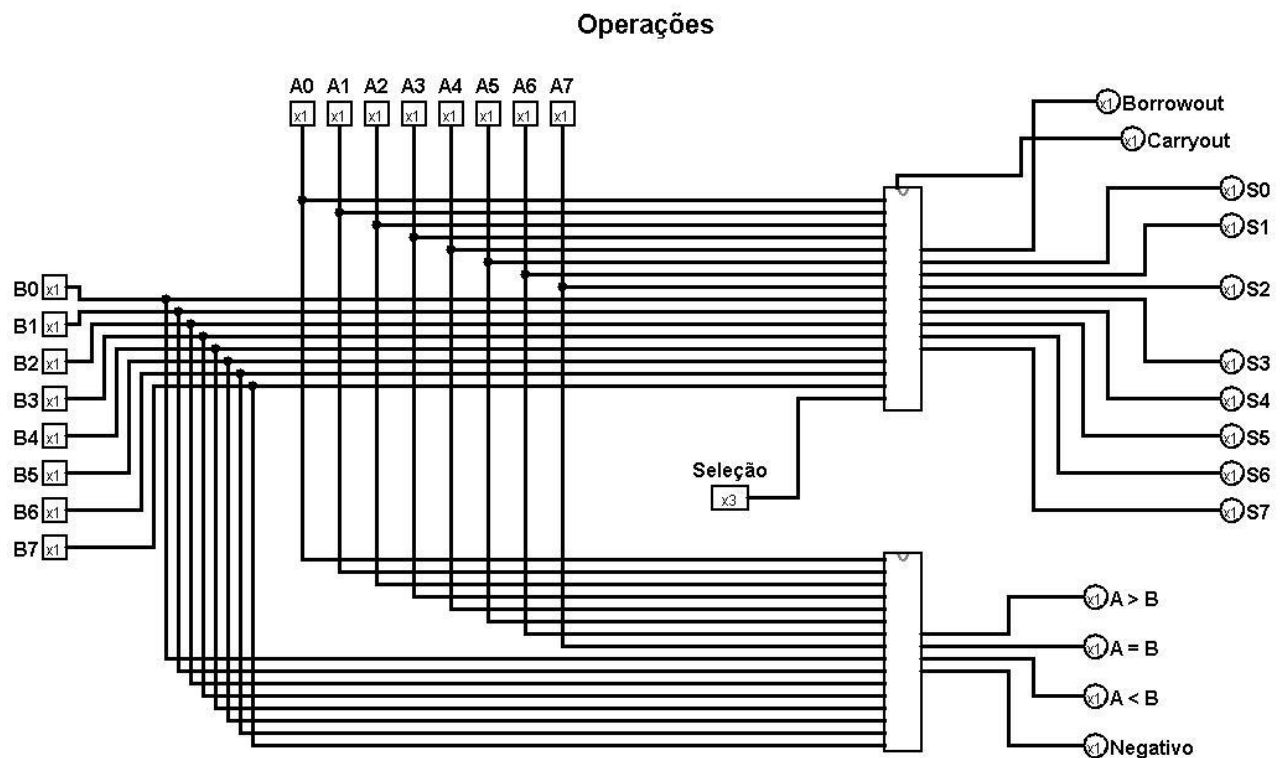


Figura 15. Circuito Operações.

Foi criado um circuito com o intuito de criar displays que mostram o resultado da operação que foi escolhida nele temos quatro entradas e sete saídas (a à g) que representam cada componente dos displays para exibir o resultado.

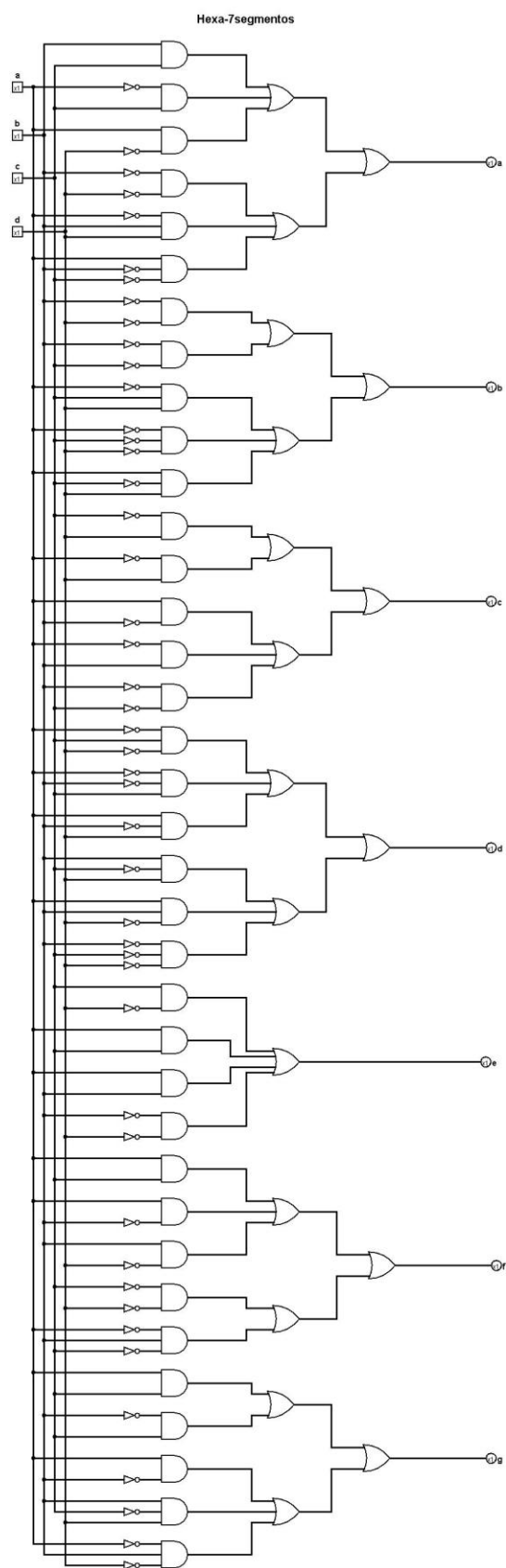


Figura 16. Circuito Hexa-7segmentos.

O último circuito que criamos foi para juntar o circuito operações e o circuito hexa-7segmentos, onde será escolhida a operação que vai ser realizada e o resultado será exibido em dois displays no formato hexadecimal.

Se as operações escolhidas for soma ou subtração o resultado será exibido no display e os sinalizadores irão mostrar se houve overflow, se as entradas são iguais ou se A é maior que B ou menor, e no caso da subtração se o resultado é negativo.

Caso a operação escolhida seja igual, maior, menor, AND ou OR os displays só irão se alterar se a condição for verdade, se for falsa os dois displays apresentarão valores 0 e se for verdade o primeiro display apresentará 8 e o segundo 0.

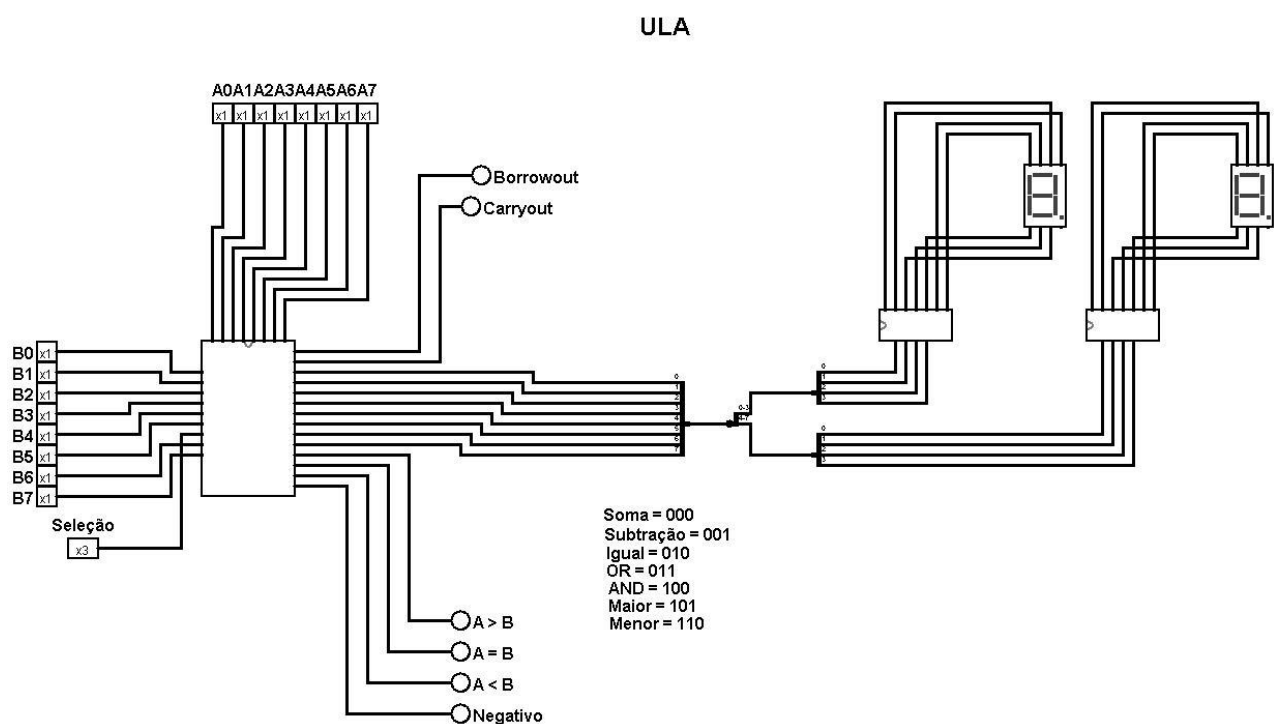


Figura 17. Circuito ULA.

Na ULA proposta a instrução possui 19bits, dividindo em opcode e operando temos: 3bits para o opcode e 16bits para o operando onde temos dois dados sendo 8bits para cada dado.

5. Conclusão

A ULA designada para esse trabalho foi projetada baseada nos circuitos digitais propostos por Langdon e Fregni (1987), Fregni e Saraiva (1995) e Tocci, Widmer e Moss (2007).

O trabalho foi dividido em módulos isolados para organizar a arquitetura interna do circuito e permitir testes separadamente com todas as operações que compõem a ULA que foi proposta.

A operações aritméticas usa a topologia de propagação vai-um e pede um isso facilitou a soma das entradas de 8bits e apresentou se na operação realizada houve overflow.

As operações de comparação, AND e OR retorna um resultado de 8bits sendo eles de valores lógicos verdadeiros ou falso.

Todas as operações foram simuladas com o auxílio do simulador Logisim testamos circuito por circuito, e todos apresentaram um funcionamento satisfatório para as operações do projeto. Através destas simulações e da execução para criar os circuitos obtivemos um melhor entendimento do que foi proposto.

6. Referências

FREGNI, Edson; SARAIVA, Antonio Mauro. Engenharia do projeto logico digital: conceitos e pratica. Sao Paulo: E. Blucher, 1995. 498p.

GÜNTZEL, José Luis. “Projeto de um Unidade Lógico-Aritmética (ULA)”. Universidade Federal de Pelotas. Pelotas, 2003.

KUDVA, Prabhakar, et al. “Synthesis of Hazard-free Customized CMOS Complex-Gate Networks Under MultipleInput Changes”. IB T.J. Watson Research Center. 1996.

LANGDON, Glen George Jr. e FREGNI, Edson. Projeto de computadores digitais. São Paulo: E. Blucher, 1987. 357p.

MANIÇOBA, Alexandre. “Projeto I - Passo a passo para a criação de um microprocessador de 4 bits. Registradores. Ula. Unidade de Controle.”, Notas de aula, Centro de Ciências Exatas e Tecnológicas da Universidade Católica de Santos, Santos, março de 2011.

MOSIS - Integrated Circuit Fabrication Service. “IBM 0.35 Micron 5HPE Process”. Disponível em: < <http://www.mosis.com/ibm/5hpe/>>. Acesso em: 10 de junho de 2011.

PATTERSON, David A.; HENNESSY, John L.. Computer organization and design: the hardware/software interface . 3. ed. Boston: Morgan Kaufmann Publishers, 2005. 621 p., ISBN 1558606041

TOCCI, Ronald J., WIDMER, N. S. e MOSS G. L. Sistemas digitais: princípios e aplicações. 10. ed. São Paulo: Prentice Hall, 2007. xxii, 804 p. ISBN 9788576050957.

RADHAKRISHNAN, Damu, “Design of CMOS circuits”, IEE Proceedings-G, Stevenage, UK. Vol. 138, n. 1, pp. 8390, fev/1991.

RADHAKRISHNAN, Damu, “Switching activity in CMOS pass networks”, IEE Electronics Letters Online, Stevenage, UK. Vol. 35, n. 24, pp. 2115-2116, Nov/1999.

RADHAKRISHNAN, Damu, WHITAKER, Sterling R. and MAKI Gary K. “Formal Design Procedures for Pass Transistor Switching Circuits”, IEEE Journal of Solid-State Circuits. Vol. SC-20, n. 2, pp. 531-536, Apr/1985.

SRINIVASAN, Chandra. “Arithmetic Logic Unit (ALU) design using reconfigurable CMOS logic”. 2003. 115 f. Tese (Mestrado de Ciências em Engenharia Elétrica) – Louisiana State University and Agricultural and Mechanical College, Louisiana, 2003.

SUZUKI, Makoto, et al. "A 1.5ns 32b CMOS ALU in Double Pass-Transistor Logic" , ISSCC 93 – IEEE International Solid-State Circuits Conference, Session 5, pp. 90-91, 1993.

STALLINGS, W. Arquitetura e organização de computadores. 8ª ed. São Paulo: Prentice Hall, 2010.