



Implementação da ULA e da Unidade de Controle

Universidade Federal de Uberlândia
Faculdade de Computação
Prof. Dr. rer. nat. Daniel D. Abdala

Na Aula Anterior ...

- O conceito de arquitetura;
- Organização Monociclo;

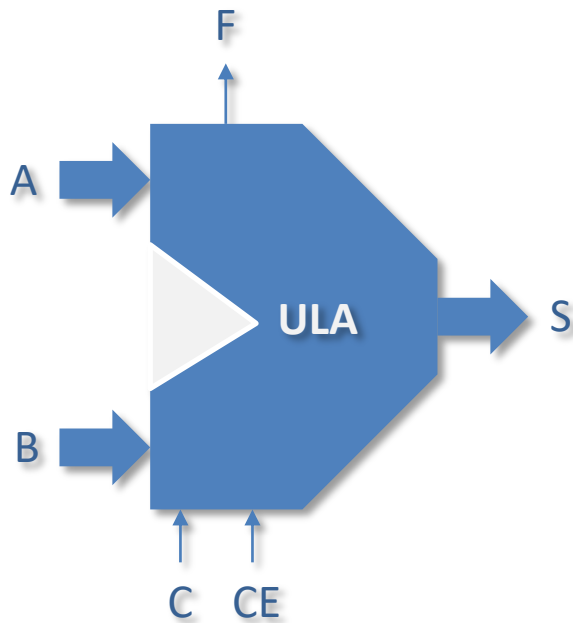
Nesta Aula

- A ULA – Unidade Lógica e Aritmética;
- Subsistemas da ULA;
- Diferentes formas de organizar a ULA;
- Subsistema de Controle;
- Sinais de Controle;

ULA – Unidade Lógica e Aritmética

- “Coração” do μ processador;
- É a coleção de subsistemas digitais que executam boa parte das funções de um processador;
- A ULA deve ser especificamente projetada para implementar um tipo de aritmética;
- $ULA-\mathbb{Z} \neq ULA-\mathbb{R}$

ULA – Visão Geral



- **A,B** – entradas de dados;
- **C** – controle, indica que operação será executada;
- **S** – saída de dados;
- **F** – flags, indicam o estado da ULA;
- **CE** – chip enable, habilita o funcionamento da ULA.

Operações Contempladas

Operações

1. Adição
2. Subtração
3. E lógico
4. OU lógico
5. NÃO-OU lógico
6. XOU lógico
7. Comparação*

Considerações

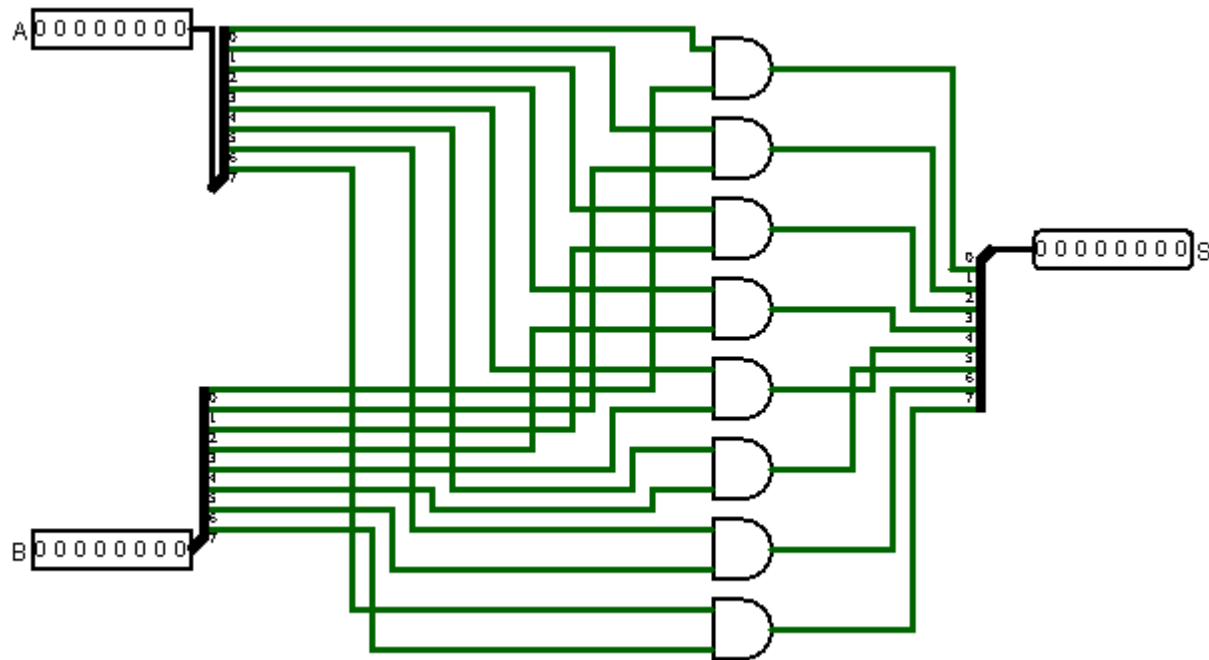
- Projetar um subsistema para cada operação;
- Executar todas as operações sempre;
- Selecionar o resultado que se deseja via multiplexação;
- Resultados das comparações são espelhados diretamente nos flags.

Flags

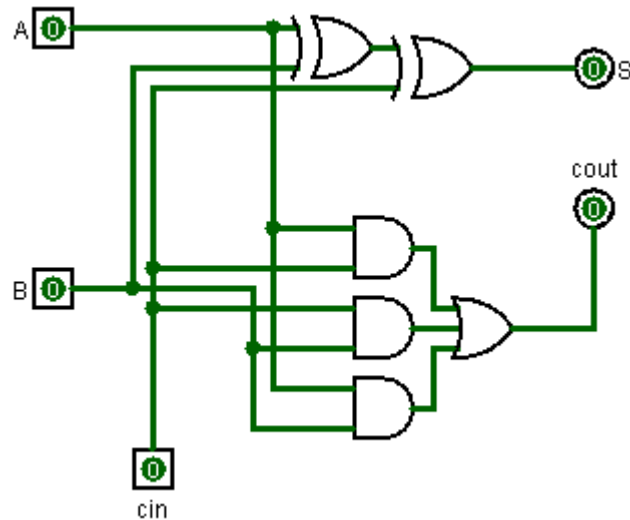


- Sinais (bits) que indicam o estado da ULA;
 1. Me $\rightarrow A < B$
 2. Ig $\rightarrow A = B$
 3. Ma $\rightarrow A > B$
 4. Io \rightarrow Integer Overflow
 5. Iu \rightarrow Integer Underflow

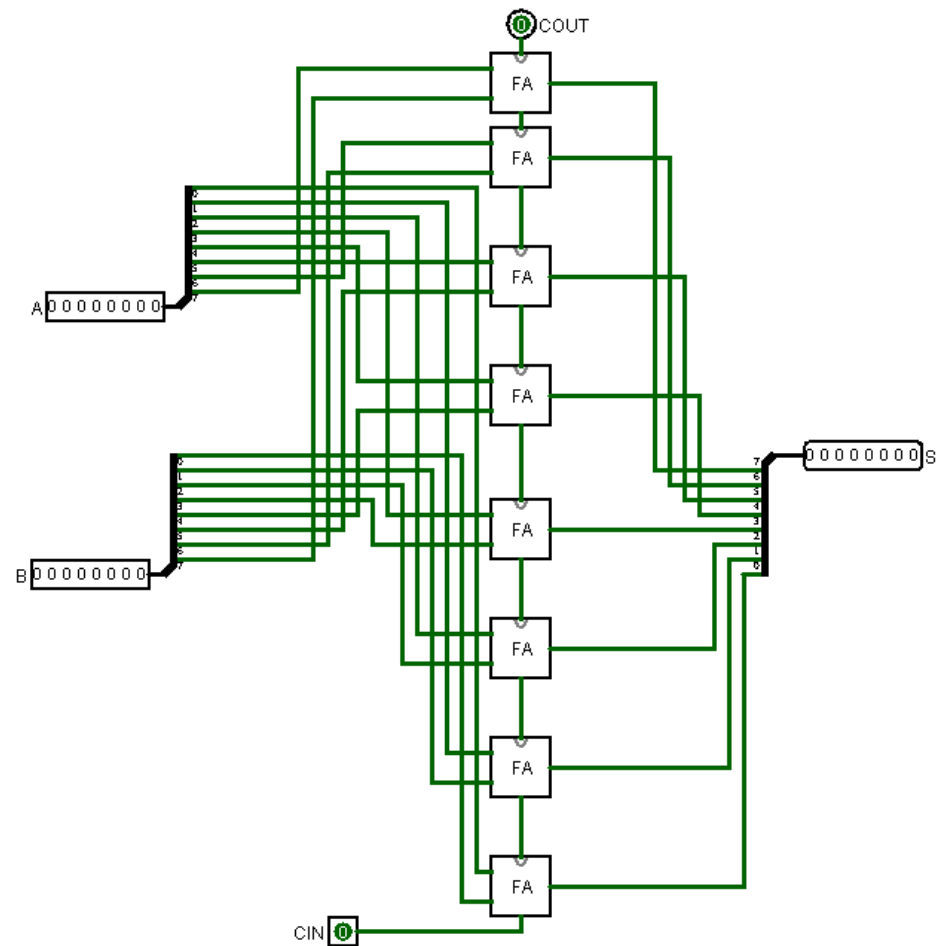
Subsistema “E” (8 bits)



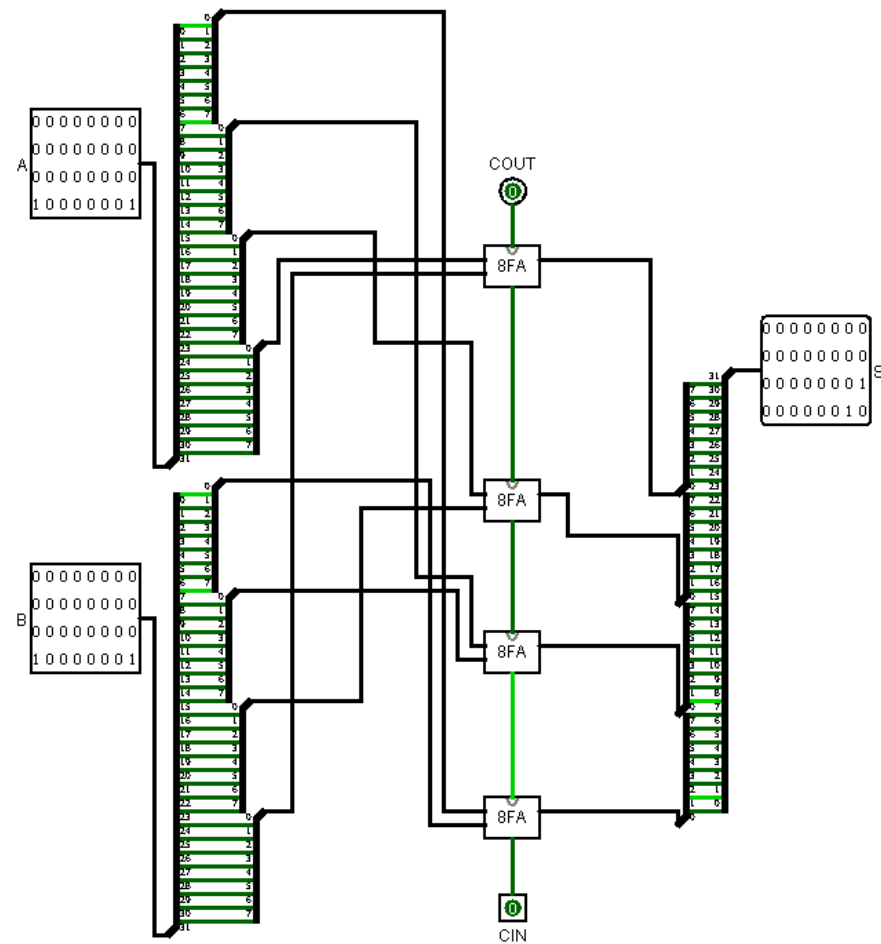
Somador Completo (1 bit)



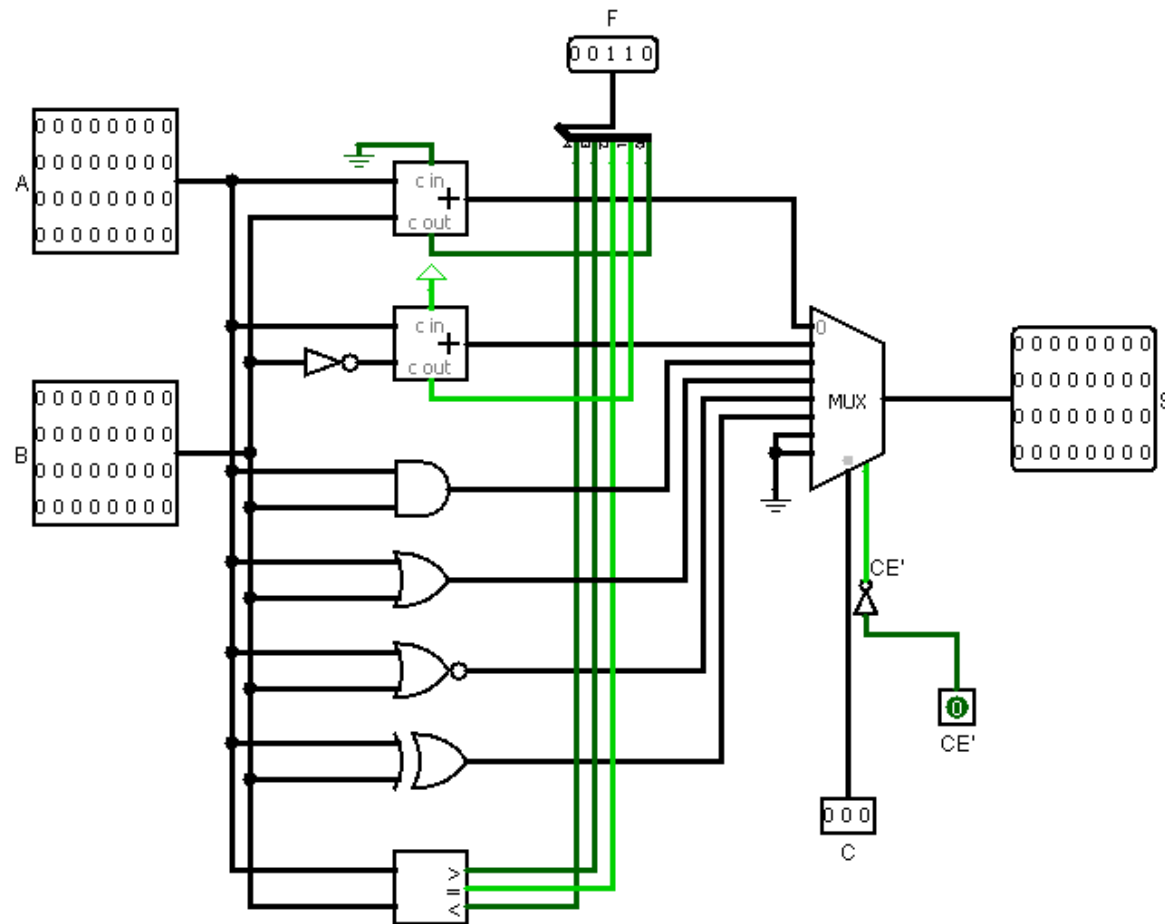
Somador (8 bits)



Somador (32 bits)



ULA – Diagrama Esquemático



Considerações ULA Apresentada

- Todas as operações são executadas paralelamente;
- Todos os circuitos consomem energia;
- Diferentes organizações da ULA podem priorizar interesses distintos;
 - Consumo de energia;
 - Velocidade de processamento;
 - Número de transistores utilizados;
 - Didática (nosso caso);

Unidade de Controle

- A função da UC é decodificar as instruções e configurar o restante da CPU para executar as instruções decodificadas;
- A decodificação converte os bits dos campos **OPCODE** e **FUNCT** para uma série de sinais de controle;
- O processo de configuração é alcançado via a definição de uma série de sinais de controle internos que roteiam a informação pelo processador;

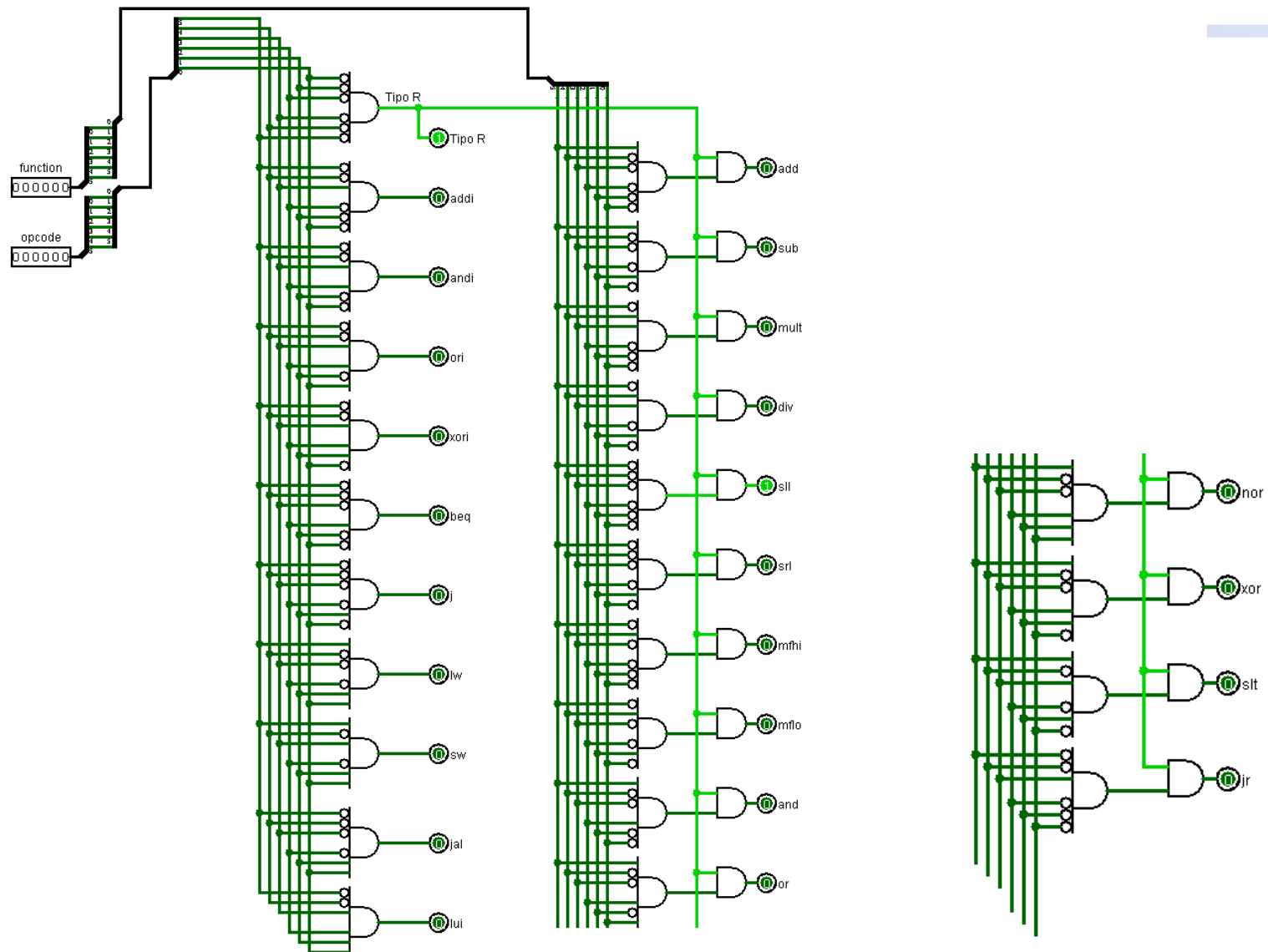
Sinais de Controle

- **RegDst** → Sinal que controla o registrador de destino de uma instrução (rd ou rt);
- **Branch** → Sinal gerado quando uma instrução de salto condicional deve ser executada;
- **MemRd** → Sinal que controla a leitura da memória;
- **MemWr** → Sinal que controla a escrita da memória
- **M2R** → Sinal que controla de a saída da ULA vai para o register file ou memória;
- **AluSrc** → Sinal que controla a fonte de dados (B) da ULA;
- **RegWr** → Sinal que controla a escrita de registradores no register file;
- **Jump** → Sinal gerado quando uma instrução de salto incondicional deve ser executada;

Tabela de Decodificação de Instruções

Instruction	op[5,4,3,2,1,0]	fc[5,4,3,2,1,0]	RegDst	Branch	MemRd	MemWr	M2R	AluSrc	RegWr	Jump
add	0 0 0 0 0 0	1 0 0 0 0 0	00	0	0	0	000	0	1	00
sub	0 0 0 0 0 0	1 0 0 0 1 0	00	0	0	0	000	0	1	00
and	0 0 0 0 0 0	1 0 0 1 0 0	00	0	0	0	000	0	1	00
or	0 0 0 0 0 0	1 0 0 1 0 1	00	0	0	0	000	0	1	00
nor	0 0 0 0 0 0	1 0 0 1 1 1	00	0	0	0	000	0	1	00
xor	0 0 0 0 0 0	1 0 0 1 1 0	00	0	0	0	000	0	1	00
slt	0 0 0 0 0 0	1 0 1 0 1 0	00	0	0	0	001	0	1	00
addi	0 0 1 0 0 0	-----	01	0	0	0	000	1	1	00
andi	0 0 1 1 0 0	-----	01	0	0	0	000	1	1	00
ori	0 0 1 1 0 1	-----	01	0	0	0	000	1	1	00
xori	0 0 1 1 1 0	-----	01	0	0	0	000	1	1	00
sll	0 0 0 0 0 0	0 0 0 0 0 0	00	0	0	0	000	0	1	00
srl	0 0 0 0 0 0	0 0 0 0 1 0	00	0	0	0	000	0	1	00
beq	0 0 0 1 0 0	-----	xx	1	0	0	xxx	0	0	00
j	0 0 0 0 1 0	-----	xx	0	0	0	xxx	x	0	01
jr	0 0 0 0 0 0	0 0 1 0 0 0	xx	0	0	0	xxx	x	0	10
jal	0 0 0 0 1 1	-----	10	0	0	0	xxx	x	0	01
lw	1 0 0 0 1 1	-----	01	0	1	0	001	1	1	00
sw	1 0 1 0 1 1	-----	xx	0	0	1	xxx	1	0	00
mul	0 0 0 0 0 0	0 1 1 0 0 0	xx	0	0	0	xxx	0	0	00
div	0 0 0 0 0 0	0 1 1 0 1 0	xx	0	0	0	xxx	0	0	00
mfo	0 0 0 0 0 0	0 1 0 0 1 0	00	0	0	0	011	0	1	00
mfhi	0 0 0 0 0 0	0 1 0 0 0 0	00	0	0	0	100	0	1	00
lui	0 0 1 1 1 1	-----	01	0	0	0	101	x	1	00

Decodificação de Instruções



Decodificação de Instruções

