



# UNIVERSIDADE PRESBITERIANA MACKENZIE

## Faculdade de Computação e Informática



### LABORATÓRIO DE ÁLGEBRA BOOLEANA E CIRCUITOS DIGITAIS

Prof. Jamil Kalil Naufal Jr.

Prof. Wallace Rodrigues de Santana

## PROJETO 2

### ULA – UNIDADE LÓGICA E ARITMÉTICA

#### 1. OBJETIVO

Construir uma Unidade Lógica e Aritmética de 8 bits que efetue as principais operações básicas presentes em uma ULA.

#### 2. APRESENTAÇÃO

Uma Unidade Lógica e Aritmética (ULA), também denominada de Arithmetic and Logic Unit (ALU), é um circuito digital usado para realizar operações lógicas e aritméticas. Ela representa o bloco funcional fundamental da Unidade Central de Processamento (CPU) de qualquer computador.

As CPUs modernas contêm ULAs muito poderosas e complexas. Além das ULAs, as CPUs modernas contêm também uma unidade de controle (UC).

A maioria das operações de uma CPU é executada por uma ou mais ULAs, que carregam dados de registros de entrada.

Um registrador é uma pequena área de armazenamento temporário disponível como parte de uma CPU.

A unidade de controle (UC) informa à ULA qual operação deve ser executada nestes dados, e a ULA armazena o resultado em um registro de saída, normalmente denominado de Acumulador. A unidade de controle move os dados entre esses registros, a ULA e a memória principal do computador.

Uma ULA pode ser dividida em duas unidades básicas:

- Unidade de Operações Aritméticas (UA) que realiza operações de:
  - Adição;
  - Subtração;
  - Transferência;
  - Incremento;
  - Decremento;



- Unidade de Operações Lógicas (UL) que realiza operações de:
  - AND;
  - OR;
  - XOR;
  - XNOR.

### 3. DESCRIÇÃO GERAL

O projeto consiste na construção de uma ULA de 8 bits composta das unidades básicas integradas:

- Unidade de Operações Aritméticas (UA): baseado no somador paralelo completo (CI 74283) para as operações de soma e subtração;
- Unidade de Operações Lógicas (UL): circuitos combinatórios baseados em circuitos integrados em portas lógicas e/ou multiplexadores;
- Multiplexador de Seleção: para a seleção de operação lógica e aritmética a ser realizada.

A atividade consiste em projetar cada uma destas unidades (Operação Lógica e Aritmética) e integrá-las através de um processo de seleção proporcionado por um Multiplexador, conforme a Figura 1:

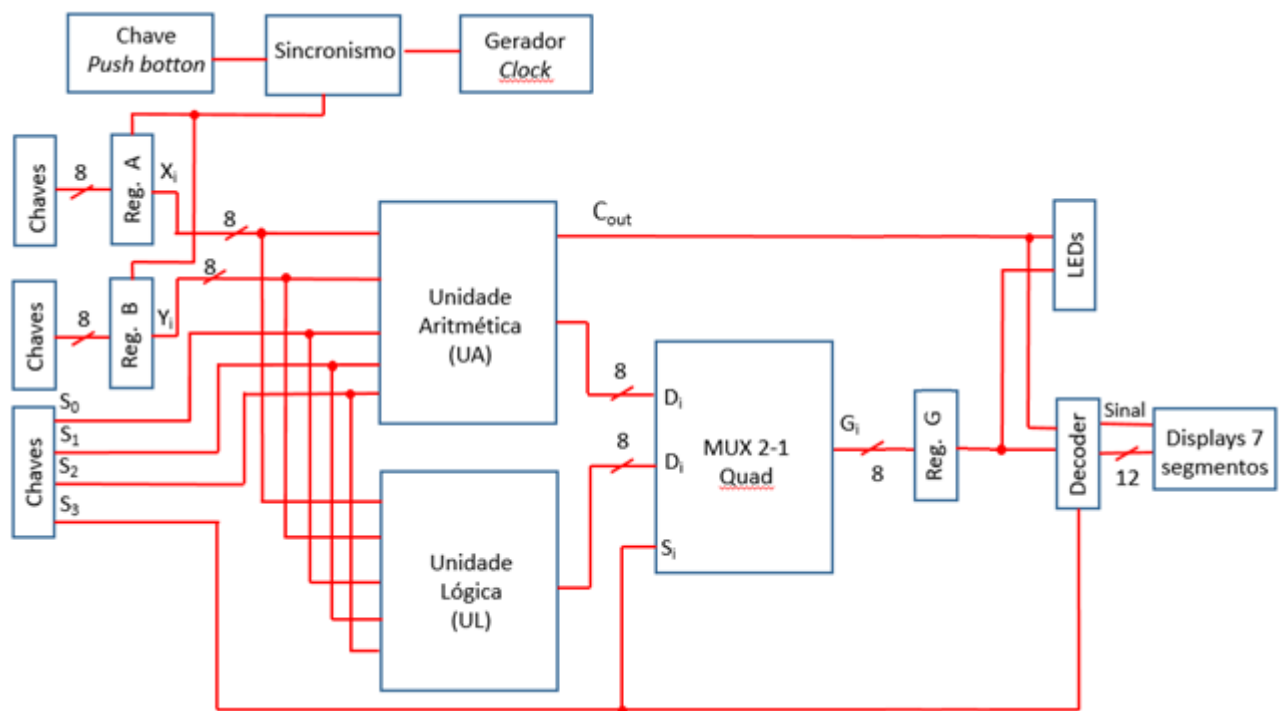


Figura 1 – Unidade Lógica e Aritmética (ULA) a ser projetada

Sendo:

- $X_i$  e  $Y_i$ : operandos de 8 bits ( $X_7, X_6, X_5, X_4, X_3, X_2, X_1$  e  $X_0$ ;  $Y_7, Y_6, Y_5, Y_4, Y_3, Y_2, Y_1$  e  $Y_0$ );
- $S_i$ : variáveis de seleção de operação ( $S_3, S_2, S_1$  e  $S_0$ );
- $G_i$ : resultado de 8 bits da operação gerada pela ULA ( $G_7, G_6, G_5, G_4, G_3, G_2, G_1$  e  $G_0$ ).



Observação:

- Quando  $S_3 = 0$ , a saída final vem da unidade aritmética;
- Quando  $S_3 = 1$ , a saída vem da unidade lógica;
- $C_{out}$  deve ser ignorado quando operações lógicas são realizadas (quando  $S_3 = 1$ ).

As Unidades de Lógica e de Aritmética compartilham as entradas de seleção  $S_1$  e  $S_0$ , mas somente a unidade aritmética usa  $S_2$ .

A ULA projetada deverá executar as seguintes operações, de acordo com os sinais de controle apresentados na Tabela 1.

Tabela 1: Operações da ULA

$S_3 S_2 S_1 S_0$	Operação	Mnemônico
0 0 0 0	$G = X$	MOV G, A
0 0 0 1	$G = X + 1$	INC G, A
0 0 1 0	$G = X - 1$	DEC G, A
0 0 1 1	$G = X + Y$	ADD G, A, B
0 1 0 0	$G = Y$	MOV G, B
0 1 0 1	$G = Y + 1$	INC G, B
0 1 1 0	$G = Y - 1$	DEC G, B
0 1 1 1	$G = X + Y' + 1$	SUB G, A, B
1 0 0 0	$G = X \text{ and } Y$	AND G, A, B
1 0 0 1	$G = X \text{ or } Y$	OR G, A, B
1 0 1 0	$G = X \text{ xor } Y$	XOR G, A, B
1 0 1 1	$G = X \text{ xnor } Y$	XNR G, A, B

Sendo:

- $S_3, S_2, S_1$  e  $S_0$ : variáveis de seleção;
- Operação: operação a ser executada em função das variáveis de seleção;
- Mnemônico: código em linguagem assembly da instrução de máquina.

O circuito somador completo (CI 74LS83) é representado no diagrama lógico da Figura 2 e será a unidade base a ser utilizada em operações aritméticas da ULA a ser projetada.

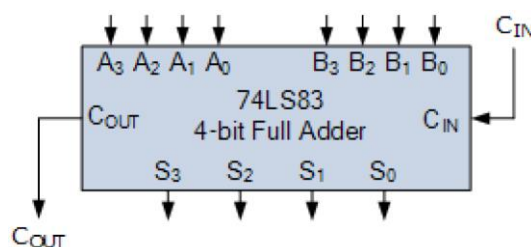


Figura 2 – Circuito lógico do somador completo (CI 74283)

Neste diagrama os terminais  $A_3, A_2, A_1$  e  $A_0$ , bem como  $B_3, B_2, B_1$  e  $B_0$ , representam cada um dos operadores,  $C_{in}$  é o Carry In (vêm um) e  $C_{out}$  é o Carry Out (vai um).



O circuito somador completo sempre calcula:

$$S = A + B + C_{in}$$

Para as chaves que alimentam os registradores A e B, deve-se utilizar o seguinte componente do Cedar ilustrado na Figura 3.

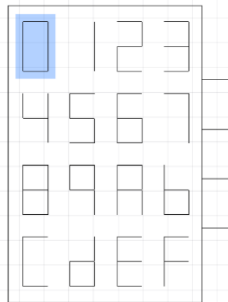


Figura 3 – 4-bit Hex Keypad

Para os operandos só devem ser aceitos os caracteres de 0 a 9. Uma sugestão é a utilização de uma memória ROM para fazer a interface entre o teclado e o registrador.



Como construir um controlador de teclado demandaria um outro projeto, para simplificar pode-se utilizar um teclado separado para cada entrada.

Para a saída, deve-se utilizar um dos seguintes componentes do Cedar ilustrados na Figura 4.

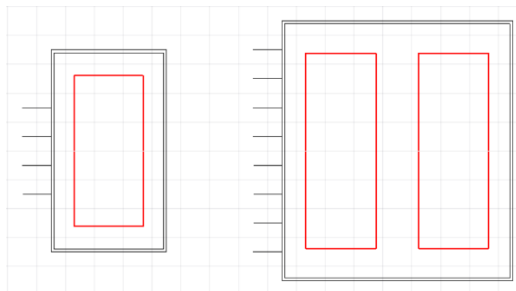


Figura 4 – 4-bit e 8-bit LED Display



Para a saída pode-se optar por utilizar um decodificador ou uma memória ROM para fazer a interface entre o registrador e o display.

#### 4. INSTRUÇÕES DE MÁQUINA (ASSEMBLY)

Instruções de máquina, código de máquina ou simplesmente *assembly* representam as instruções que um processador é capaz de executar. Imagine o seguinte trecho de código escrito em uma linguagem de alto nível como C++, Java ou Python, por exemplo:



**Código 1 – linguagem de alto nível**

```
001  (...)
002      byte i=5; //Declara e inicializa a variável i=5
003      byte j=3; //Declara e inicializa a variável j=3
004      byte k;   //Declara a variável k
005      k=i+j;    //Armazena em k o resultado de i+j
006      i++;      //Incrementa a variável i, ou seja, i=i+1
007      j--;      //Decrementa a variável j, ou seja, j=j-1
008      k=i-j;    //Armazena em k o resultado de i-j
009  (...)

```

Após o processo de compilação, as instruções de alto nível são convertidas para instruções de máquina ou *assembly*. As variáveis *i*, *j* e *k* são endereços de memória que contém, respectivamente, os valores 5, 3 e 0. O tipo *byte* indica que estas variáveis possuem oito bits de tamanho, ou seja, armazenam valores decimais de 0 a 255, que é a capacidade de processamento da ULA deste projeto. Quando este código for executado, os valores contidos nos endereços de memória são carregados nos registradores disponíveis no processador.

**Código 2 – linguagem de baixo nível (*assembly*)**

```
001  (...)
002      MOV A, 5;
003      MOV B, 3;
004      ADD G, A, B;
005      INC G, A;
006      DEC G, B;
007      SUB G, A, B;
008  (...)

```



As instruções *assembly* que carregam os valores contidos em um endereço de memória para um registrador não serão implementados neste projeto. As instruções das linhas 002 e 003 servem apenas para ilustrar o mecanismo de carregamento de dados em um registrador. Diz-se que estas são instruções que carregam um valor imediato no registrador.

Note que a ULA deste projeto só é capaz de realizar operações sobre registradores, faltando, portanto, instruções que carregam dados da memória (como *LOAD*) e que gravam dados na memória (como *STOR*). Por isso, o código *assembly* listado não seria executado adequadamente, pois o valor da soma armazenado no Reg. G (linha 004) seria sobrescrito pela instrução de incremento da linha seguinte (linha 005).

O estudo mais aprofundado dos componentes de um computador e a programação em linguagem *assembly* serão abordados na disciplina Organização de Computadores do terceiro semestre (ou terceira etapa).



## 5. ELABORAÇÃO DO PROJETO

O projeto de circuito digital da Unidade Lógica e Aritmética deverá conter os seguintes componentes:

### 5.1. Unidade Aritmética

Projetar a Unidade Aritmética no CEDAR utilizando o somador completo de 4 bits compatível com o CI 74283, disponível no CEDAR em “6 - Add & Compare”. Utilizar multiplexadores para selecionar as operações desejadas. Como resposta a essa questão, apresentar a captura de tela (printscreen) da montagem no CEDAR e adicionar no Relatório Técnico um exemplo de teste para cada operação, conforme as operações aritméticas apresentadas nas Tabelas 1 e 2. Utilizar uma ou mais áreas de trabalho do CEDAR para criar a Unidade Aritmética, não esquecendo de identificá-las.

Tabela 2: Operações da Unidade Aritmética (UA)

Código de seleção $S_2 S_1 S_0$	Operação desejada $G(A + B + C_{in})$	Entradas necessárias ao somador		
		A	B	$C_{in}$
0 0 0	X (transferir)	0000 0000	X	0
0 0 1	$X + 1$ (incrementar)	0000 0000	X	1
0 1 0	$X - 1$ (decrementar)	1111 1111	X	0
0 1 1	$X + Y$ (adicionar)	Y	X	0
1 0 0	Y (transferir)	0000 0000	Y	0
1 0 1	$Y + 1$ (incrementar)	0000 0000	Y	1
1 1 0	$Y - 1$ (decrementar)	1111 1111	Y	0
1 1 1	$X + Y' + 1$ (subtração por complemento de 2)	$Y'$	X	1



Os circuitos somador e subtrator deverão compartilhar os mesmos componentes. Implementar o somador separado do subtrator acarretará desconto de 1,0 ponto na nota.

### 5.2. Unidade Lógica

Projetar a Unidade Lógica (AND, OR, XOR e XNOR) no CEDAR utilizando portas lógicas e minimizando o circuito lógico combinacional para obter a menor representação possível, caso seja necessário. Utilizar multiplexadores para selecionar as operações desejadas. Como resposta a essa questão, apresentar a captura de tela (printscreen) da montagem no CEDAR e adicionar no Relatório Técnico um exemplo de teste para cada operação, conforme as operações lógicas apresentadas na Tabela 1. Utilizar uma ou mais áreas de trabalho do CEDAR para criar a Unidade Lógica, não esquecendo de identificá-las.

### 5.3. Integração entre as Unidades Lógica e Aritmética

Combinar as unidades aritmética e lógica de modo que as entradas X e Y e a saída S sejam independentes da realização de uma operação, seja ela qual for o tipo, e de acordo com o diagrama lógico da Figura 1. Em outras palavras, as entradas X e Y devem alimentar simultaneamente as unidades aritmética e lógica, que terão uma saída S única. Utilizar multiplexadores para selecionar as operações desejadas. Como resposta a essa questão, apresentar a captura de tela (printscreen) da montagem no CEDAR e adicionar no Relatório Técnico um exemplo de teste para cada operação, conforme as operações aritméticas e lógicas apresentadas na Tabela



1. Utilizar uma ou mais áreas de trabalho do CEDAR para criar esta integração, não esquecendo de identificá-las.

#### 5.4. Registradores, sincronismo e decodificador

A seguir serão destacados os blocos funcionais adicionais ao projeto de ULA.

##### 5.4.1. Registradores

Os registradores Reg. A e Reg. B armazenam os valores dos operandos da entrada ( $X_i$  e  $Y_i$ ) gerados a partir do teclado, enquanto o registrador Reg. G armazena o valor da saída. A Figura 5 apresenta uma unidade básica de registradores. Esta estrutura armazena o bit de cada operando registrador através de flip-flops tipo D sensível a borda de subida.

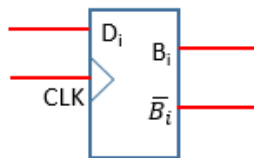


Figura 5 – Unidade básica sugerida para armazenar cada bit dos operandos

O clock pode ser utilizado como entrada dos registrados para sincronização de armazenamento dos operandos.

##### 5.4.2. Sincronismo

O gerador de clock gera um sinal digital (onda quadrada) de 100 Hz e é utilizado para sincronizar o armazenamento dos dados dos registradores A, B e G (Reg. A, Reg. B e Reg. G) a partir do acionamento da chave push-button.

O bloco de sincronismo deve, a partir de um pulso gerado pela chave push-button ou do sinal de 100 Hz, gerar um pulso para sincronizar o carregamento do conteúdo dos operandos ( $X_i$  e  $Y_i$ ) a partir do teclado para os registradores A e B (Reg. A e Reg. B) e do resultado para o registrador da saída G (Reg. G).

##### 5.4.3. Decodificador

A ULA possui duas saídas, LEDs e display de 7 segmentos com resultados em decimal. As operações aritméticas devem ser apresentadas tanto na saída com LEDs quanto na saída com os displays de 7 segmentos (em formato decimal com resultados das operações entre 0 a 255). As operações lógicas devem ser realizadas somente na saída com LEDs.

O bloco decodificador deverá receber o valor das operações aritméticas e permitir a sua apresentação no formato BCD em 7 segmentos (formato decimal). Observar que se a operação aritmética for negativa, ela deverá ser informada, ou seja, deverá haver uma sinalização pelo LED  $C_{out}$  e sinalização complementar no display de 7 segmentos para indicar resultado negativo.

Os LEDs informados na Figura 2 representam os resultados das operações em binário puro, devendo seu valor ser interpretado em função das operações lógicas ou aritméticas realizadas pela ULA.



#### 5.4.3.1. Decodificação de binário para BCD

Observar que a decodificação deverá ser detalhada de forma que ela possa apresentar nos três displays de 7 segmentos todas as somas e subtrações em decimal utilizando 8 bits no formato decimal.

Na Figura 6 é apresentada uma forma de decodificação que faz a detecção e correção para apresentar as somas em formato BCD acima de 9 em decimal.

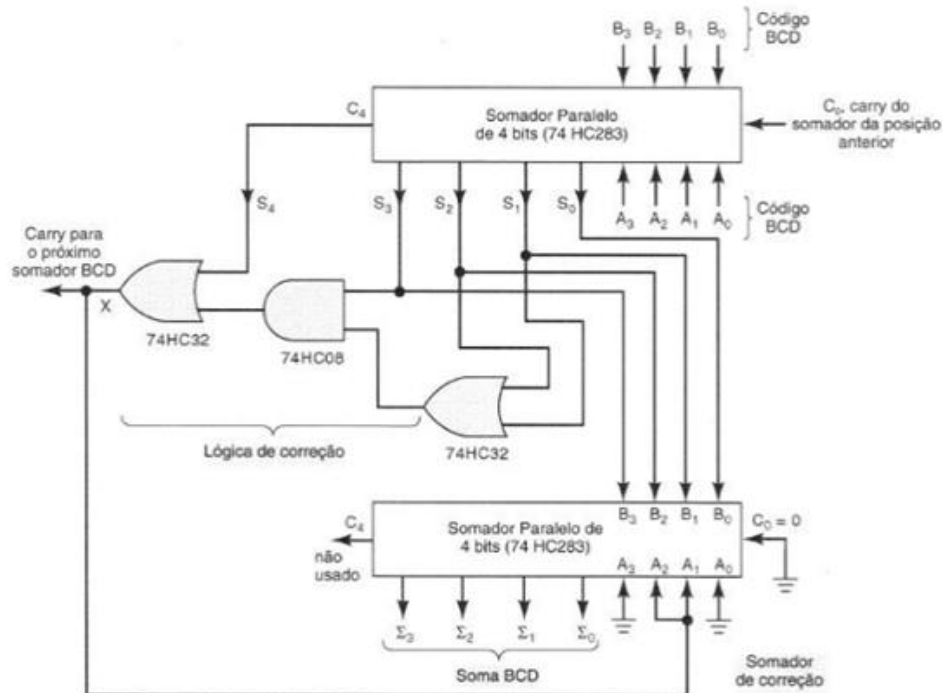


Figura 6 – Exemplo de somador BCD para somas maiores que 9

Observar que são utilizados dois somadores de quatro bits e um circuito detector para correção das somas binárias acima de 9.

A Figura 7 apresenta um exemplo da soma em BCD com correção.

$$\begin{array}{rcl} 0110 & \leftarrow & \text{BCD para 6} \\ + 0111 & \leftarrow & \text{BCD para 7} \\ \hline 1101 & \leftarrow & \text{soma inválida} \\ 0110 & \leftarrow & \text{soma 6 para corrigir} \\ \hline 0001 & & 0011 \\ \hline 1 & & 3 \end{array} \leftarrow \text{BCD para 13}$$

Figura 7 – Exemplo de soma BCD com correção

Observar que a soma de 0110 + 0111 ultrapassa o valor binário BCD, portanto é realizada uma correção somando-se adicional 0110, de tal forma a obter-se o valor correto em BCD, ou seja, 0001 0011.





Notar que as saídas corrigidas da soma agora podem ser as entradas de um decodificador de 7 segmentos, tal como o CI CD4511.

Observar que este exemplo não permite que todas as operações de soma em 4 bits sejam convertidas para BCD de forma correta. Igualmente não contempla as operações de subtração em 4 bits.

Um dos desafios do projeto é conseguir realizar todas as operações de soma e subtração com decodificação correta (BCD) para que os três displays de 7 segmentos apresentem os resultados das operações em decimal.

Notar que no CEDAR o display de 7 segmentos já inclui um decodificador, mas ele não faz a decodificação em BCD e sim em hexadecimal.

Opcionalmente, o grupo poderá substituir o decodificador por um módulo de memória ROM para fazer a interface com o display de 7 segmentos.

## **6. ENTREGAS DO PROJETO**

### **6.1. Documentação**

A entrega de um Relatório Técnico contendo todas as etapas do projeto presentes neste documento e desenvolvido nos mesmos moldes dos relatórios realizados nos laboratórios. O mesmo deverá ser entregue obrigatoriamente somente via Moodle em data definida pelo professor.

A entrega deverá também incluir:

- Relatório técnico (PDF) contendo os itens solicitados no template;
- Arquivo CDL do projeto gerado pelo CEDAR.

Portanto, são dois arquivos de entrega. Não entregue compactado. Faça o upload individualmente de cada um deles.

### **6.2. Apresentação**

A apresentação em sala de aula será realizada através da apresentação da montagem realizada no CEDAR. A demonstração do funcionamento do circuito deverá permitir observar de forma clara todas as diferentes saídas em função das entradas definidas. Um aspecto importante é a organização da montagem no CEDAR de forma a permitir entender de forma fácil e clara a operação do sistema.

**OBS.:** a não apresentação em sala de aula acarretará nota zero no projeto, ainda que seja entregue o relatório com o descritivo técnico.

Para a simulação dos testes, poderá ser utilizada a Tabela 3 contendo um exemplo de teste para cada operação contida nas Tabelas 1 e 2. Apresentar e explicar no relatório como os testes foram realizados e os resultados obtidos com base em imagens e explicações no CEDAR. Observar que são doze operações ao todo.



Tabela 3: Simulação dos testes de validação da ULA

S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>	Operação	X <sub>i</sub>	Y <sub>i</sub>	LEDs	DISPLAY
		X <sub>7</sub> X <sub>6</sub> X <sub>5</sub> X <sub>4</sub> X <sub>3</sub> X <sub>2</sub> X <sub>1</sub> X <sub>0</sub>	Y <sub>7</sub> Y <sub>6</sub> Y <sub>5</sub> Y <sub>4</sub> Y <sub>3</sub> Y <sub>2</sub> Y <sub>1</sub> Y <sub>0</sub>	C <sub>out</sub> G <sub>7</sub> G <sub>6</sub> G <sub>5</sub> G <sub>4</sub> G <sub>3</sub> G <sub>2</sub> G <sub>1</sub> G <sub>0</sub>	

## 7. DATAS DE ENTREGA

### 7.1. Documentação (Relatório Técnico)

Data de entrega definida pelo professor.

### 7.2. Apresentação

Data de apresentação definida pelo professor.

## 8. CRITÉRIOS DE AVALIAÇÃO DO PROJETO

### 8.1. Relatório Técnico: até 40% da nota

Contendo as etapas do projeto elencados no Descritivo Técnico, sendo:

- Correta formatação do relatório, conforme template disponibilizado (1,0 ponto);
- Referências bibliográficas no formato ABNT (1,0 ponto);
- Conter todas as simulações (2,0 pontos).

### 8.2. Montagem no CEDAR: até 60% da nota

Montagem realizada de forma organizada e demonstração de todas as simulações, sendo:

- Correto funcionamento da Unidade Aritmética (2,0 pontos);
- Correto funcionamento da Unidade Lógica (2,0 pontos);
- Correto funcionamento dos teclados, LEDs, displays e sinais de sincronismo (2,0 pontos).



Os circuitos somador e subtrator deverão compartilhar os mesmos componentes. Implementar o somador separado do subtrator acarretará desconto de 1,0 ponto na nota.

--FIM--