

Prova Escrita
Área: Sistemas Digitais e Arquitetura de Computadores

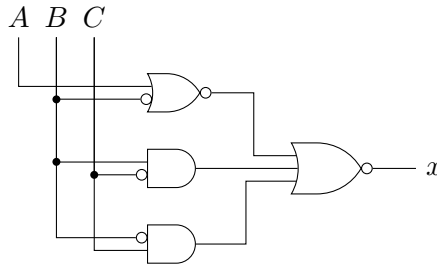
GABARITO

- 1 -) (2,0) Usando apenas as instruções, pseudo-instruções e chamadas de sistema do MIPS32 listadas no Anexo I, implemente um programa em linguagem de montagem (*assembly*) do MIPS32 que conte quantas vezes um dígito especificado d aparece em um número inteiro n . A entrada é composta por duas linhas: a primeira contém o número n e a segunda o dígito d . Estão garantidos os seguintes limites: $0 < n < 2^{31}$ e $0 \leq d \leq 9$. Sua solução deve realizar a operação utilizando divisão inteira. Considere um processador sem *branch delay slot*.

Conforme os métodos apresentados no Capítulo 2 e Apêndice A de PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: a interface hardware-software. 5. ed. Rio de Janeiro: Elsevier, 2017.

```
1      .text
2      .globl main
3
4 main:
5      # ler n
6      li $v0, 5
7      syscall
8      move $t0, $v0      # t0 = n
9
10     # ler d
11     li $v0, 5
12     syscall
13     move $t1, $v0      # t1 = d
14
15     # inicializar contador
16     li $t2, 0          # t2 = contador
17
18 loop:
19     beq $t0, $zero, fim # se n == 0, fim
20
21     # dividir n por 10
22     li $t3, 10
23     div $t0, $t3
24
25     mfhi $t4            # resto (último dígito)
26     mflo $t0            # quociente (n = n/10)
27
28     # comparar dígito extraído com d
29     bne $t4, $t1, loop
30     addi $t2, $t2, 1    # contador++
31
32     j loop
33
34 fim:
35     # imprimir resultado
36     li $v0, 1
37     move $a0, $t2
38     syscall
39
40     # encerrar
41     li $v0, 10
42     syscall
```

- 2 -) (2,0) Utilizando teoremas da Álgebra de Boole, encontre uma expressão booleana mínima na forma normal disjuntiva (FND) equivalente ao circuito lógico abaixo. Apresente o **circuito lógico** correspondente à **expressão booleana mínima**, bem como todos os passos de redução da expressão booleana. Identifique o teorema aplicado em cada passo; você pode aplicar em um único passo o mesmo teorema para vários termos da expressão. Utilize notação algébrica de circuitos digitais (por exemplo, $F = \overline{A} B + C$).



Conforme métodos apresentados no Capítulo 4 de TOCCI, R.J.; WIDMER, N.S; MOSS, G.L. Sistemas digitais: princípios e aplicações. 12^a ed, Pearson Prentice-Hall, 2019.:

$$\overline{\overline{A + \overline{B} + B\overline{C} + \overline{B} C}}$$

$$\overline{\overline{A + \overline{B}} \overline{B\overline{C}} \overline{\overline{B} C}} \quad \text{DeMorgan}$$

$$(A + \overline{B}) \overline{B\overline{C}} \overline{\overline{B} C} \quad \text{Dupla Negação}$$

$$(A + \overline{B}) (\overline{B} + \overline{\overline{C}}) (\overline{\overline{B}} + \overline{C}) \quad \text{DeMorgan}$$

$$(A + \overline{B}) (\overline{B} + C) (B + \overline{C}) \quad \text{Dupla Negação}$$

$$(A\overline{B} + AC + \overline{B}\overline{B} + \overline{B}C)(B + \overline{C}) \quad \text{Distributiva}$$

$$(A\overline{B} + AC + \overline{B} + \overline{B}C)(B + \overline{C}) \quad \text{Multiplicação lógica}$$

$$(A\overline{B} + AC + \overline{B})(B + \overline{C}) \quad \text{Multivariado / Absorção}$$

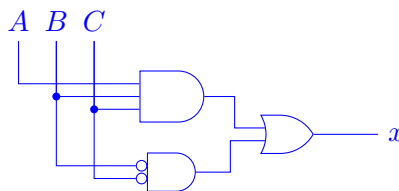
$$A\overline{B}B + ACB + \overline{B}B + A\overline{B}\overline{C} + AC\overline{C} + \overline{B}\overline{C} \quad \text{Distributiva}$$

$$ACB + A\overline{B}\overline{C} + \overline{B}\overline{C} \quad \text{Multiplicação lógica / Complemento}$$

$$ACB + \overline{B}\overline{C} \quad \text{Multivariado / Absorção}$$

Resposta final: $ABC + \overline{B}\overline{C}$

Circuito:



3 -) (2,0) Utilizando Mapas de *Karnaugh*, encontre um circuito lógico mínimo, que realize as tabelas verdade abaixo. Apresente ainda o **circuito lógico mínimo** e sua **expressão booleana**, bem como o Mapa de Karnaugh.

#	A	B	C	D	X
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	1	0	1
4	0	0	1	1	0
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	0
8	0	1	1	1	1
9	1	0	0	0	1
10	1	0	0	1	0
11	1	0	1	0	1
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	1
15	1	1	1	0	0
16	1	1	1	1	1

#	A	B	C	D	X
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	1
11	1	0	1	0	0
12	1	0	1	1	1
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

Conforme os métodos apresentados no Capítulo 4 de TOCCI, R.J.; WIDMER, N.S; MOSS, G.L. Sistemas digitais: princípios e aplicações. 12ª ed, Pearson Prentice-Hall, 2019.:

		C	0	0	1	1
		D	0	1	1	0
A	B					
0	0		1	0	0	1
0	1		0	1	1	0
1	1		0	1	1	0
1	0		1	0	0	1

$\overline{B} \overline{D} + B D$

		C	0	0	1	1
		D	0	1	1	0
A	B					
0	0		1	1	1	1
0	1		0	0	0	0
1	1		0	0	0	0
1	0		0	1	1	0

$\overline{A} \overline{B} + \overline{B} D$

- 4 -) (2,0) Os valores a seguir correspondem à representação hexadecimal de um valor em ponto flutuante (*float*) no padrão IEEE 754 de 32 bits (sendo o bit mais significativo para o sinal, os próximos 8 bits para o expoente — armazenado com *bias* — e os demais 23 bits menos significativos para a mantissa/fração/*significand*). Converta cada representação hexadecimal de *float* para o respectivo valor decimal.

(a) C2288000₁₆

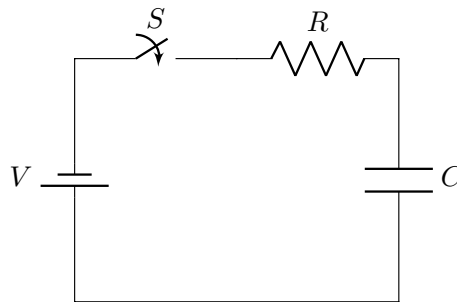
(b) 3F200000₁₆

Conforme os métodos apresentados no Capítulo 3 de PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: a interface hardware-software. 5. ed. Rio de Janeiro: Elsevier, 2017; e no Capítulo 2 de TOCCI, R.J.; WIDMER, N.S; MOSS, G.L. Sistemas digitais: princípios e aplicações. 12^a ed, Pearson Prentice-Hall, 2019:

(a) -42.125

(b) 0.625

- 5 -) (1,0) No circuito abaixo a chave S é fechada no instante $t = 0$, começando o **carregamento** de um capacitor C inicialmente descarregado. A capacitância do capacitor é $C = 18 \mu\text{F}$ e a resistência do resistor é $R = 33 \Omega$. Determine o instante em que a **diferença de potencial entre os terminais do resistor** é igual à **diferença de potencial entre os terminais do capacitor**.



Conforme os métodos apresentados nos Capítulos 5 e 10 de BOYLESTAD, R. L. Introdução à Análise de Circuitos. 12a ed. Pearson Prentice-Hall, São Paulo, 2012

$$V_c(t) = V(1 - e^{-t/(RC)})$$

$$V_r(t) = V e^{-t/(RC)}$$

$$V_c(t) = V_r(t)$$

$$V(e^{-t/(RC)}) = V(1 - e^{-t/(RC)})$$

$$e^{-t/(RC)} = 1 - e^{-t/(RC)}$$

$$2e^{-t/(RC)} = 1$$

$$e^{-t/(RC)} = 1/2$$

$$-t/(RC) = \ln(1/2) = -\ln(2)$$

$$t = RC \ln(2)$$

$$t = 33 \times 18 \times 10^{-6} \times \ln(2)$$

$$t = 0.4117 \text{ ms} = 411.7 \mu\text{s}$$

- 6 -) (1,0) Um valor de saída de 6,8V proveniente de um sensor analógico é utilizado como entrada em um conversor Analógico-Digital. Utilizado o método de aproximações sucessivas:

a) Codifique o sinal para um registro de seis bits com limite de escala de 10 V.

b) Visto que há um erro intrínseco no processo de conversão, qual o valor de tensão será expresso de fato?

Conforme métodos apresentados no Capítulo 6 de GROOVER, Mikell P. Automação industrial e sistemas de manufatura. 3. ed. São Paulo: Pearson Education do Brasil, 2011. 581 p. ISBN 9788576058717 (broch.).

Respostas

a) 101011

b) 6,719V

a) Solução

Primeira tensão de referência $\rightarrow 10/2 = 5,0V$

$6,8 > 5,0 \rightarrow 1$ Tensão maior que a de referência

$6,8 - 5,0 = 1,8$

Próxima tensão de referência $\rightarrow 5,0/2 = 2,5V$

$1,8 < 2,5 \rightarrow 0$ Tensão menor que a de referência

Próxima tensão de referência $\rightarrow 2,5/2 = 1,25 V$

$1,8 > 1,25 \rightarrow 1$ Tensão maior que a de referência

$1,8 - 1,25 = 0,55$

Próxima tensão de referência $\rightarrow 0,625 = 1,25/2$

$0,55 < 0,625 \rightarrow 0$ Tensão menor que a de referência.

Próxima tensão de referência $\rightarrow 0,3125 = 0,625/2$

$0,55 > 0,313 \rightarrow 1$ Tensão maior que a de referência.

$0,55 - 0,313 = 0,2375$

Próxima tensão de referência $\rightarrow 0,15625 = 0,3125/2$

$0,2375 > 0,156 \rightarrow 1$ Tensão maior que a de referência.

b) Solução

$1 \times 5,0 V = 5V$

$0 \times 2,5V = 0V$

$1 \times 1,25V = 1,25V$

$0 \times 0,625V = 0V$

$1 \times 0,3125V = 0,3125V$

$1 \times 0,15625V = 0,15625V$

$5,0 + 0,0 + 1,25 + 0 + 0,3125 + 0,15625 = 6,71875V$

Arredonda para 3 casas decimais $\Rightarrow 6,719V$

ANEXO I

Legenda de Operandos para as Instruções

label, target: qualquer rótulo textual

\$t1, \$t2, \$t3: qualquer registrador inteiro

-100: inteiro de 16 bits com sinal (*signed*)

100: inteiro de 16 bits sem sinal (*unsigned*)

100000: inteiro de 16 bits com sinal (*signed*)

-100(\$t2): soma do inteiro de 16 bits com sinal (*signed*) com o conteúdo de \$t2

Lista de registradores inteiros do MIPS32

\$zero: sempre contém o valor 0

\$at: reservador para o montador

\$v0, \$v1: registradores de valores de retorno de funções

\$a0, \$a1, \$a2, \$a3: registradores de passagem de argumentos

\$t0 a \$t9: registradores temporários (não preservados em chamadas de função)

\$s0 a \$s7: registradores salvos (preservados em chamadas de função)

\$k0, \$k1: reservados para o sistema operacional

\$gp: registrador global pointer

\$sp: registrador stack pointer

\$fp: registrador frame pointer

\$ra: registrador de retorno de função

HI e LO são registradores para a multiplicação e divisão (acessados por instruções como **mfhi**, **mflo**, **mthi** e **mtlo**)

Instruções Assembly do MIPS32

add \$t1, \$t2, \$t3: adição com overflow, faz $\$t1 = \$t2 + \$t3$

addi \$t1, \$t2, -100: adição de imediato com overflow, faz $\$t1 = \$t2 + \text{imediato de 16 bits com sinal}$

sub \$t1, \$t2, \$t3: subtração com overflow, faz $\$t1 = \$t2 - \$t3$

addu, addiu e subu: versão sem overflow de add, addi e sub, respectivamente

and \$t1, \$t2, \$t3: E lógico bit-a-bit, faz $\$t1 = \$t2 \text{ E-lógico bit-a-bit } \$t3$

andi \$t1, \$t2, 100: E lógico bit-a-bit, faz $\$t1 = \$t2 \text{ E-lógico bit-a-bit com imediato de 16 bits estendido com zeros}$

or \$t1, \$t2, \$t3: OU lógico bit-a-bit, faz $\$t1 = \$t2 \text{ OU-lógico bit-a-bit com } \$t3$

ori \$t1, \$t2, 100: OU lógico bit-a-bit, faz $\$t1 = \$t2 \text{ OU-lógico bit-a-bit com imediato de 16 bits estendido com zeros}$

beq \$t1, \$t2, label: Desvie se igual. Desvia para label se \$t1 igual a \$t2

bne \$t1, \$t2, label: Desvie se diferente. Desvia para label se \$t1 diferente de \$t2

j target: Desvio incondicional. Desvia para o endereço target (um rótulo)

div \$t1, \$t2: Divisão inteira com overflow. Divide \$t1 por \$t2, coloca o resultado da divisão inteira em LO e o resto em HI

divu: versão sem overflow de div

mfhi \$t1: move o conteúdo do registrador HI para \$t1

mflo \$t1: move o conteúdo do registrador LO para \$t1

mthi \$t1: move o conteúdo de \$t1 para o registrador HI

mtlo \$t1: move o conteúdo de \$t1 para o registrador LO

syscall: executa uma chamada de sistema conforme o código especificado em \$v0.

lw \$t1, -100(\$t1): Carrega palavra de 4 bytes. Define \$t1 com o conteúdo do endereço efetivo de memória correspondente.

sw \$t1, -100(\$t1): Armazena palavra. Armazena o conteúdo de \$t1 no endereço efetivo de memória correspondente.

pseudo-instruções Assembly do MIPS32

`li $t1, -100$`: atribui um valor de 16-bits ao registrador \$t1, estendido com sinal
`li $t1, 100$`: atribui um valor de 16-bits ao registrador \$t1, estendido com zero
`li $t1, 100000$`: atribui um valor de 32-bits ao registrador \$t1
`move $t1, $t2$`: copia para \$t1 o conteúdo de \$t2

Códigos de chamada de sistema do MIPS32 (syscall)

1: imprime na saída padrão o inteiro em `$a0`
5: lê da entrada padrão um valor inteiro, armazenando em `$v0`
10: encerra o programa

Diretivas

`.ascii` : armazena strings no segmento de dados, sem adicionar o terminador nulo
`.asciiz` : armazena strings no segmento de dados, adiciona o terminador nulo
`.data` : itens subsequentes são armazenados no setor de dados no próximo endereço disponível
`.text` : itens subsequentes (instruções) são armazenados no setor text (de instruções) no próximo endereço disponível
`.globl`: Declara o(s) rótulo(s) listado(s) como globais para permitir que sejam referenciados a partir de outros arquivos.

Exemplo

```
1      .text
2      .globl main
3
4 main:
5      # ler um inteiro
6      li $v0, 5
7      syscall
8      move $t0, $v0
9
10 loop:
11      beq $t0, $zero, fim
12
13      # imprimir um inteiro
14      li $v0, 1
15      move $a0, $t0
16      syscall
17
18      addi $t0, $t0, -1 # decrementa
19
20      j loop
21
22 fim:
23      # encerrar
24      li $v0, 10
25      syscall
```



Assinaturas do documento



Código para verificação: **4I8Z84ZG**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:



YURI KASZUBOWSKI LOPES (CPF: 063.XXX.259-XX) em 24/11/2025 às 09:00:56

Emitido por: "SGP-e", emitido em 02/06/2021 - 13:17:43 e válido até 02/06/2121 - 13:17:43.

(Assinatura do sistema)



RAFAEL RODRIGUES OBELHEIRO (CPF: 788.XXX.800-XX) em 24/11/2025 às 09:12:48

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:41:02 e válido até 30/03/2118 - 12:41:02.

(Assinatura do sistema)



ROBERTO SILVIO UBERTINO ROSSO JUNIOR (CPF: 283.XXX.604-XX) em 24/11/2025 às 13:44:01

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:37:29 e válido até 30/03/2118 - 12:37:29.

(Assinatura do sistema)



RICARDO FERREIRA MARTINS (CPF: 628.XXX.906-XX) em 24/11/2025 às 13:47:18

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:43:00 e válido até 30/03/2118 - 12:43:00.

(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTlwMjJfMDAwNDY1NTZfNDY1ODdfMjAyNV80SThaODRaRw==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00046556/2025** e o código **4I8Z84ZG** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.