

CI INOVADOR

**PROCESSO SELETIVO DE INGRESSO
NO PROGRAMA DE RESIDÊNCIA EM
MICROELETRÔNICA – CI BRASIL INOVAÇÃO 2**

PROVA DE CONTEÚDO DIGITAL

Execução



UFSC

Coordenação



Softex

Realização

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



01

Boolean equations use variables that can assume only two possible values: TRUE or FALSE. For this reason, they are especially useful in digital logic, where circuits operate by distinguishing between two logical states.

Boolean equations are particularly appropriate for describing digital systems because they

- (A) represent information through two logical states, matching the binary nature of digital circuits.
- (B) allow variables to assume any continuous value between TRUE and FALSE.
- (C) replace the need for logic gates in all digital systems.
- (D) describe only analog signals rather than digital behavior.
- (E) represent values through only one logical state, matching the non-binary nature of digital circuits.

02

In Boolean algebra, the complement of a variable represents its inverse. A variable written without a bar is called the true form, while the same variable with a bar over it is called the complementary form.

“True form” in this context means that the variable

- (A) is written without the TRUE mark, not necessarily that its value has a complementary mark.
- (B) always has the logical value TRUE in every circuit.
- (C) cannot be inverted in a Boolean equation.
- (D) is written without a complement mark, not necessarily that its value is TRUE.
- (E) is more important than its complementary form.

03

Boolean equations follow a specific order of operations. NOT has the highest precedence, followed by AND, and then OR. This order determines how expressions are interpreted when parentheses are not explicitly shown.

How should the expression $Y = A + BC$ be correctly interpreted?

- (A) $Y = (B \text{ AND } C) \text{ AND } A$
- (B) $Y = A \text{ OR } (B \text{ AND } C)$
- (C) $Y = (A \text{ OR } B) \text{ AND } C$
- (D) $Y = A \text{ AND } (B \text{ OR } C)$
- (E) $Y = \text{NOT } A \text{ OR } (B \text{ AND } C)$

04

In Boolean equations, products are performed before sums. Therefore, when an expression contains both AND and OR operations, the AND operation must be evaluated first unless parentheses indicate otherwise.

What is the main function of operator precedence in Boolean equations?

- (A) To produce ambiguity in the role of logical operations.
- (B) To ensure that OR operations are always calculated before AND operations.
- (C) To prevent ambiguity in the interpretation of logical expressions.
- (D) To eliminate the need for variables and complements.
- (E) To make Boolean equations identical to decimal arithmetic.

Anotações



05

Boolean terminology helps designers describe digital logic precisely. Terms such as literal, product, implicant, minterm, sum, and maxterm identify the structure and role of expressions inside Boolean equations.

What is the main pedagogical purpose of defining these terms before writing Boolean equations from truth tables?

- (A) To settle a pedagogical purpose for teaching how to write Boolean equations.
- (B) To show that Boolean algebra is unrelated to digital circuit design.
- (C) To replace truth tables with ordinary arithmetic calculations.
- (D) To prove that all Boolean expressions must contain only three variables.
- (E) To establish a precise vocabulary for interpreting and constructing logic functions.

06

Karnaugh maps are a graphical method used to simplify Boolean equations. Unlike purely algebraic manipulation, which can accidentally produce a different equation if done carelessly, K-maps organize combinable terms next to each other in a grid. This visual arrangement helps identify terms such as PA and $P\bar{A}$, which can be combined to eliminate the variable A .

Based on the text, what is the main advantage of using Karnaugh maps in Boolean minimization?

- (A) They help to point out equations that cannot be combined or simplified.
- (B) They make it easier to visually identify adjacent terms that can be combined and simplified.
- (C) They allow Boolean equations with any number of variables to be simplified without limits.
- (D) They eliminate the need to understand implicants or complementary variables.
- (E) They always produce a completely different equation from the original one.

07

Assembly language is described as the human-readable representation of a computer's native language. Each instruction identifies both the operation to be performed and the operands involved. In the MIPS architecture, these operands may include registers, memory, and constants.

Considering the information presented in the text, what can be inferred about the structure of an assembly language instruction?

- (A) It replaces the computer's native language with a completely unrelated human language.
- (B) It changes the performed operation and the operands involved in that operation.
- (C) It specifies both the operation to be executed and the operands involved in that operation.
- (D) It stores all program data permanently inside the processor's arithmetic unit.
- (E) It eliminates the need for registers, memory, and constants in computer architecture.

08

The concept of the MOSFET was proposed by Julius Edgar Lilienfeld in 1925, long before the MOS transistor was successfully fabricated in the 1960s. The main obstacle was the oxide-silicon interface, which initially contained many surface states. These surface states trapped charge carriers and caused poor conduction. Later advances in semiconductor technology and clean-room fabrication allowed the oxide to be grown on silicon with almost no surface states, resulting in high transconductance.

Considering the technological challenge described in the text, what can be inferred about the delayed fabrication of the MOS transistor?

- (A) The delay happened because of the over-control of oxide-silicon interface.
- (B) The delay occurred because the theoretical concept of the MOSFET had not yet been proposed.
- (C) The delay was caused by the complete absence of interest in semiconductor devices before the 1960s.
- (D) The delay happened because clean rooms increased the number of surface states in the oxide layer.
- (E) The delay was mainly caused by fabrication limitations related to controlling the oxide-silicon interface.

09

Amplifiers and ADCs are examples of analog functions, circuits that **must** process each point on a waveform (e.g., a voice signal) with great care to avoid effects such as noise and “distortion.”

While Kirchoff’s laws **can** always be utilized to solve any circuit, the Thevenin and Norton theorems can both simplify the algebra and, more importantly, provide additional insight into the operation of a circuit.

Analyze the modalization of the two modal verbs in bold in the excerpts above and select the alternative that correctly describes the role of the highlighted elements.

- (A) **Must** indicates recommendation for analog circuits that should process waveform in a direct way; **can** expresses vulnerability, pointing out that Kirchhoff’s laws are sensitive for evaluation purposes.
- (B) **Must** indicates that analog circuits are physically unable to process waveform points; **can** expresses a contrast between algebraic simplification and circuit operation.
- (C) **Must** expresses a future prediction about analog circuits; **can** indicates prohibition against using Kirchhoff’s laws.
- (D) **Must** expresses a weak suggestion that analog circuits may process waveforms; **can** indicates obligation to use Kirchhoff’s laws in every circuit.
- (E) **Must** expresses necessity or obligation, indicating that analog circuits are required to process every point on a waveform carefully; **can** expresses possibility, showing that Kirchhoff’s laws are available for solving any circuit.

10

A good circuit designer analyzes and understands the circuit before giving **it** to the computer for a more accurate analysis. A bad circuit designer, on the other hand, allows the computer to think for **him/her**.

Analyze the referents of the pronouns bolded in the text and select the alternative that correctly describes the role of the highlighted elements:

- (A) “it” refers to the designer, expressing the accurate analysis made by the computer; “him/her” refers to the computer’s bad circuit, showing that the computer could not replace the designer’s actions.
- (B) “it” refers to the computer, indicating that the computer is analyzed by the circuit designer; “him/her” refers to the circuit, suggesting that the circuit performs the analysis.
- (C) “it” refers to a more accurate analysis, showing what the designer gives to the circuit; “him/her” refers to the computer, indicating that the designer thinks for the machine.
- (D) “it” refers to the circuit, indicating what is given to the computer for more accurate analysis; “him/her” refers to a bad circuit designer, showing that the computer is being allowed to replace the designer’s own reasoning.
- (E) “it” refers to the small-signal model, although it is not mentioned in the sentence; “him/her” refers to the good circuit designer, showing that good designers allow the computer to think for them.

Anotações



11

Verifique se nas seguintes 5 operações aritméticas a seguir ocorre *overflow* usando 6 bits. Os números estão representados em complemento de dois.

Assinale **SIM** para as ocorrências de *overflow* e **NÃO** para os casos onde não há estouro de capacidade.

1. () 111111b + 110001b
2. () 100000b + 011111b
3. () 101010b + 010101b
4. () 001111b - 010101b
5. () 110000b - 001111b

A sequência correta é

- (A) SIM - SIM - SIM - SIM - SIM.
- (B) SIM - NÃO - NÃO - NÃO - NÃO.
- (C) NÃO - NÃO - NÃO - NÃO - NÃO.
- (D) NÃO - NÃO - SIM - NÃO - SIM.
- (E) NÃO - SIM - SIM - SIM - SIM.

12

Uma função $F(A,B,C,D)$, com A sendo a entrada mais significativa, é definida pela soma de min-terms $F = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$, sem condições "don't care".

Com base nessas informações, considere as afirmativas a seguir.

I $\rightarrow$ A forma mínima em soma de produtos é $F = B' \cdot D' + B \cdot D$.

II $\rightarrow$ A função equivale à operação XNOR entre B e D, isto é, $F = 1$ quando B e D são iguais.

III $\rightarrow$ A função depende das quatro variáveis A, B, C e D.

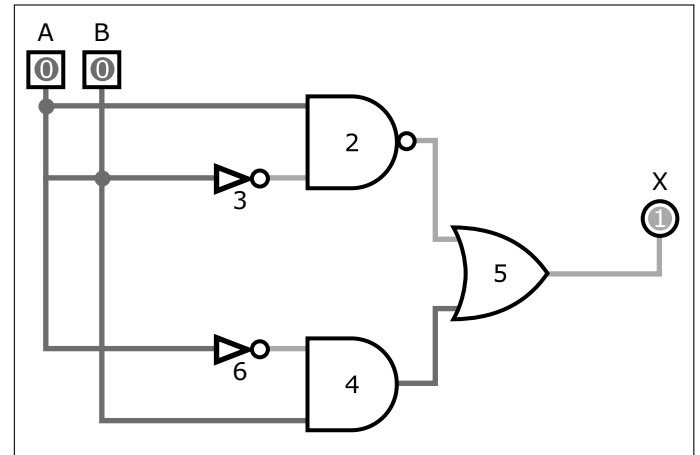
IV $\rightarrow$ A função, em produto de somas, pode ser escrita como $F = (B + D') \cdot (B' + D)$.

Estão corretas

- (A) apenas I e II.
- (B) apenas I e III.
- (C) apenas II e IV.
- (D) apenas III e IV.
- (E) apenas I, II e IV.

13

O esquemático mostrado a seguir é construído por portas lógicas operadas com álgebra booleana. Este circuito tem entradas A e B e saída X determinada pela relação lógica entre as portas lógicas com as quais é construído.



Sobre o circuito apresentado e as portas lógicas com as quais ele é construído, assinale V (verdadeiro) ou F (falso) em cada alternativa a seguir.

- () As portas lógicas 3 e 6 terão sempre o mesmo nível lógico na saída, por executarem a mesma operação lógica.
- () A porta lógica indicada por 5 tem a saída em nível lógico baixo se e somente se as duas entradas dessa porta lógica têm esse mesmo nível lógico simultaneamente.
- () Considerando a entrada A fixada em nível lógico alto, a saída da porta lógica 4 será dependente do nível lógico instantâneo da entrada B.

A sequência correta é

- (A) F - V - F.
- (B) F - F - F.
- (C) F - V - V.
- (D) V - F - V.
- (E) V - V - V.

Anotações



14

Considere a tabela de correspondências a seguir, que apresenta uma contagem de 4 dígitos em base decimal, em base binária natural e em código Gray.

Decimal	Binário	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

A respeito do código Gray, assinale a alternativa correta.

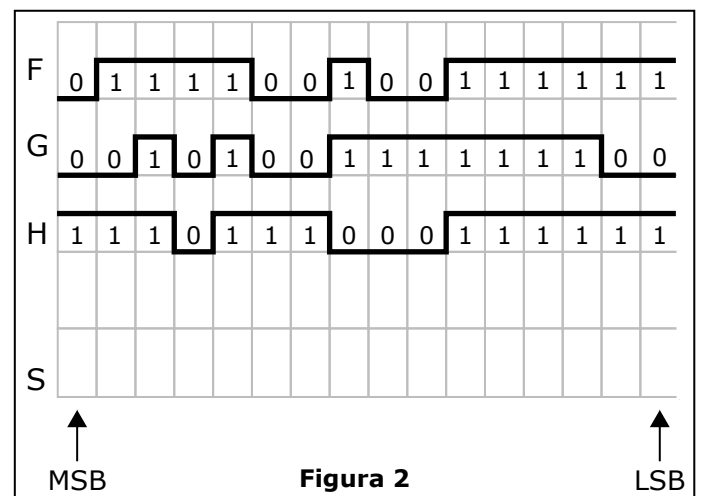
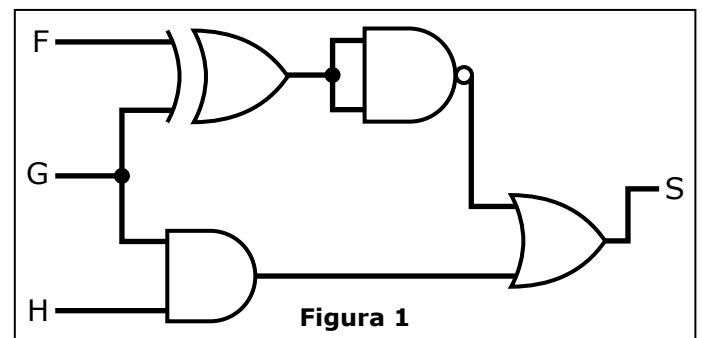
- (A) Garante que dois bits tenham seus níveis lógicos trocados na transição entre quaisquer duas palavras adjacentes na contagem.
- (B) Diferencia-se do binário natural pela presença de redundância.
- (C) Foi desenvolvido principalmente para aumentar a capacidade de armazenamento de memória digital.
- (D) É um recurso exclusivo para implementação de dispositivos e componentes digitais eletromecânicos como encoders rotativos.
- (E) Existem, em um código Gray de “n” bits, exatamente 2n combinações distintas.

15

Considere o esquemático (Figura 1) e o comportamento dinâmico da saída “S” (Figura 2), o qual apresenta a variação das entradas “F”, “G” e “H”.

Analisar o esquemático (Figura 1) e complete o comportamento dinâmico da saída “S” na Figura 2.

A cada um dos dezesseis intervalos de tempo na Figura 2, atribua um bit igual a “0” ou igual a “1” para a saída “S”, de acordo com o valor lógico calculado em função das entradas e do esquemático apresentado. Considere MSB = bit + significativo e LSB = bit - significativo.



Assinale a alternativa que apresenta o valor em hexadecimal correspondente ao padrão binário obtido para a saída “S”.

- (A) BF4C
- (B) 9B33
- (C) 7E2C
- (D) AF3C
- (E) 8B33

16

Dada a seguinte equação $Y = \overline{A}C + \overline{A}B + A\overline{B}C + BC$, a alternativa que a representa na forma canônica compacta por soma de produtos é

- (A) $Y = \Sigma(2,4,5,7)$.
- (B) $Y = \Sigma(1,2,3,5,7)$.
- (C) $Y = \Sigma(1,3,4,6,7)$.
- (D) $Y = \Sigma(2,3,4,6,7)$.
- (E) $Y = \Sigma(1,3,5,6)$.

17

Considerando a álgebra booleana, suas propriedades e o Teorema de DeMorgan, assinale a alternativa correta.

- (A) $A + A = 2.A$
- (B) $A(A + B) = A+B$
- (C) $!A + !B = !(A + B)$
- (D) $A + 1 = 1$
- (E) $!A = -A$

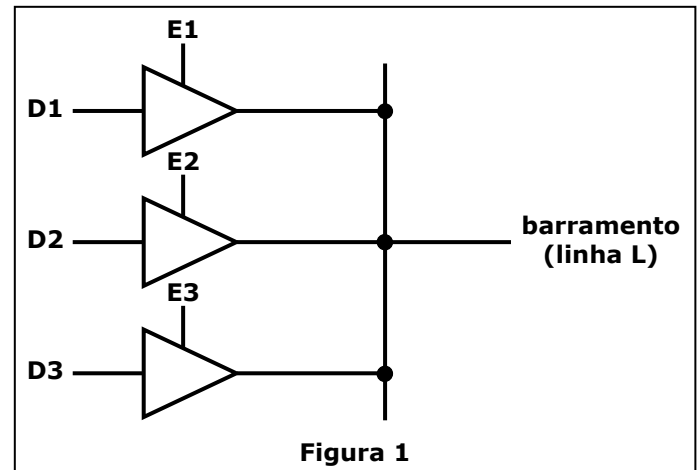
18

Sobre mecanismos de *checksum*, síndrome e erros em sistemas digitais, assinale a alternativa correta.

- (A) Códigos de síndrome maximizam a quantidade de informação útil armazenada em um dispositivo de memória volátil.
- (B) Uma soma de checagem (*checksum*) garante a detecção de todos os erros que possam ocorrer durante a transmissão de dados.
- (C) O mecanismo de FEC (Forward Error Correction) exige comunicação bidirecional para operar o mecanismo de reenvio de informações corrompidas.
- (D) Sistemas protegidos por paridade com 1 bit têm a capacidade de detectar a ocorrência de mudanças indesejadas em qualquer número ímpar de bits na palavra.
- (E) O *checksum* e a paridade têm a mesma capacidade de detecção de erros, diferindo apenas na quantidade de bits de redundância utilizada.

19

Três buffers tri-state compartilham a mesma linha de um barramento interno, conforme a Figura 1. Cada buffer i ($i=1,2$ e 3) conduz seu dado D_i para a linha somente quando seu sinal de habilitação E_i vale 1 (habilitação ativa no nível alto); com $E_i = 0$, a saída fica em alta impedância (Z).



A Tabela 1 mostra os sinais aplicados em cinco intervalos de tempo (o símbolo X indica dado irrelevante, pois o respectivo buffer está em Z).

Intervalo	E1	D1	E2	D2	E3	D3
t1	1	0	0	X	0	X
t2	0	X	1	1	0	1
t3	0	X	0	X	0	X
t4	1	1	1	0	0	X
t5	0	X	1	0	0	X

Em qual intervalo o barramento assume o nível lógico alto (H), de forma estável e sem conflito?

- (A) t1
- (B) t2
- (C) t3
- (D) t4
- (E) t5

20

No dimensionamento de blocos combinacionais de um caminho de dados, é preciso relacionar o número de entradas e saídas de multiplexadores, decodificadores e codificadores.

Considere as afirmativas a seguir.

I → Um decodificador 3:8 possui 3 entradas e 8 saídas.

II → Um codificador binário com 16 entradas requer 5 saídas.

III → Um multiplexador necessita de 5 entradas de seleção para selecionar uma entre 32 fontes de dados.

IV → Um multiplexador $2^n:1$ implementa qualquer função de n variáveis quando elas são ligadas às entradas de seleção e cada entrada de dados recebe uma constante (0 ou 1).

Está(ão) correta(s)

- Ⓐ apenas I.
- Ⓑ apenas II.
- Ⓒ apenas I, III e IV.
- Ⓓ apenas II, III e IV.
- Ⓔ I, II, III e IV.

Anotações

21

O seguinte trecho de código escrito em assembly MIPS é executado em um sistema baseado no processador projetado por Harris e Harris, cujo espaço de endereçamento é mapeado em memória, tanto para RAM quanto para dispositivos periféricos (E/S mapeado em memória):

```
lui $s0, 0x7FFF
```

```
lw $t0, 0xF000($s0)
```

Considere os imediatos em complemento de dois (signed 16-bit), sabendo que, nessa arquitetura específica, o espaço reservado para a memória RAM principal engloba os endereços de 0x00000000 até 0x7FFFFFFF; que o intervalo de 0x80000000 em diante é estritamente reservado para registradores de controle de periféricos de Entrada/Saída (E/S); e que não há exceção de proteção de memória, sendo o endereço apenas interpretado pelo barramento. Observe que o imediato de 16 bits em sinal deve ser estendido para 32 bits antes da soma.

De acordo com as instruções dadas, o código realiza

- (A) um acesso inválido à memória RAM, resultando em um erro de overflow em tempo de execução, pois tenta carregar um dado a partir do endereço relativo contido em \$s0 modificado pelo imediato.
- (B) uma leitura legítima da memória RAM principal, pois a instrução lui carrega a metade superior do endereço, e a instrução lw aplica um deslocamento com extensão de sinal, que resulta no endereço 0x7FFFFFF0.
- (C) uma operação de Entrada/Saída (E/S), pois a instrução lui define a base alta do endereço e, ao somar o imediato 0xF000 com extensão de sinal no lw, o endereço efetivo resultante é 0x7FEEF000, acessando a região de controle.
- (D) uma leitura legítima da memória RAM principal, pois, devido à extensão de sinal do imediato de 16 bits na instrução lw, o endereço efetivo calculado é 0x7FEEF000.
- (E) uma operação de Entrada/Saída (E/S), pois, devido à extensão de sinal do imediato 0xF000 (cujo bit mais significativo é 1), o endereço final calculado é 0x8000EF00, caindo na faixa reservada para periféricos.

22

No livro *Digital Design and Computer Architecture* (2ª edição), os autores David e Sarah Harris discutem como a Arquitetura do Conjunto de Instruções (ISA) do MIPS adota uma filosofia de design *Reduced Instruction Set Computer* (RISC). Uma das características fundamentais dessa filosofia é a simplificação dos modos de endereçamento para otimizar o caminho de dados e o tempo de ciclo do processador.

Assinale a alternativa que descreve corretamente as limitações e os modos de endereçamento nativos do MIPS.

- (A) O MIPS suporta o modo de endereçamento memória-memória direto por meio de instruções unificadas, permitindo somar o conteúdo de dois endereços da RAM e salvar em um terceiro endereço de memória sem usar registradores.
- (B) A instrução lw \$t0, (\$t1, \$t2) ilustra o modo de endereçamento por registrador indexado, sendo o endereço final na memória RAM calculado pela soma direta dos valores de dois registradores.
- (C) O modo de endereçamento imediato permite que uma instrução acesse qualquer endereço de 32 bits da memória RAM diretamente, pois o valor do endereço completo é embutido dentro dos 16 bits inferiores da própria instrução.
- (D) O MIPS é uma arquitetura do tipo *load/store*, o que significa que as operações aritméticas e lógicas só podem ser realizadas entre registradores, sendo o endereçamento por base mais deslocamento (ou indexado por imediato) o único modo de endereçamento de dados suportado nativamente na memória RAM.
- (E) Instruções de desvio condicional, como beq \$t0, \$t1, label, utilizam o modo de endereçamento pseudo-direto, pois os 26 bits do imediato substituem completamente os bits inferiores do *Program Counter* (PC).

Anotações



23

A Arquitetura ISA funciona como um contrato entre o programador de assembly e o projetista do hardware. Quando simplificada (mantendo instruções de tamanho fixo, poucos modos de endereçamento e operações estritamente registrador-registrador), uma ISA permite que os engenheiros projetem microarquiteturas muito mais otimizadas.

Considerando os conceitos de caminho de dados (*datapath*), controle e *pipelining*, assinale a alternativa que explica corretamente o porquê e o como uma arquitetura simplificada resulta em uma microarquitetura mais eficiente.

- (A) A uniformidade no tamanho e formato das instruções simplifica drasticamente o estágio de busca (*fetch*) e decodificação, permitindo localizar rapidamente os campos dos registradores devido à posição fixa desses campos no formato da instrução, o que facilita a implementação de um *pipeline* profundo e de alta frequência.
- (B) Instruções simplificadas e de tamanho fixo eliminam a necessidade de uma memória Cache de instruções, permitindo que o processador busque o código diretamente da memória RAM principal sem penalidades de ciclo de clock.
- (C) A existência de arquitetura simplificada obriga o uso de uma Unidade de Controle puramente microprogramada em ROM, o que reduz a área de silício necessária para os circuitos combinacionais do processador, em comparação com o controle por lógica armada (*hardwired*).
- (D) A microarquitetura é forçada a unificar os barramentos de dados e instruções (Arquitetura Von Neumann pura) ao limitar o acesso à memória RAM exclusivamente às instruções de *load* e *store*, eliminando, assim, os conflitos de hardware (*structural hazards*) no *pipeline*.
- (E) A simplificação das instruções permite que o caminho de dados execute qualquer instrução aritmética complexa em um único ciclo de clock extremamente curto, eliminando a necessidade de técnicas de previsão de desvio (*branch prediction*).

24

O fluxo de execução sequencial de um programa pode ser alterado utilizando instruções de desvio. Na arquitetura MIPS, essas instruções são categorizadas principalmente em desvios condicionais (*branches*) e desvios incondicionais (*jumps*), operando com formatos de instrução distintos (Tipo-I e Tipo-J) e modos de endereçamento específicos.

Considerando o comportamento dessas instruções no hardware do processador, assinale a alternativa que descreve corretamente as características e restrições dos tipos de desvio.

- (A) A instrução de desvio condicional *beq* utiliza o formato Tipo-J, o que lhe permite saltar para qualquer endereço absoluto de 32 bits na memória de instruções, desde que a condição de igualdade seja satisfeita.
- (B) O desvio incondicional *j* (*Jump*) calcula seu endereço alvo somando o imediato de 26 bits diretamente ao conteúdo atual do registrador *Program Counter* (PC), o que caracteriza o modo de endereçamento relativo ao PC.
- (C) As instruções de desvio condicional, como *beq* e *bne*, utilizam o endereçamento relativo ao PC, pois o imediato de 16 bits contido na instrução representa um deslocamento em palavras (*words*), permitindo saltar para frente ou para trás dentro de um limite de aproximadamente 128 KB a partir do PC atual.
- (D) A instrução *jr* (*Jump Register*) é um desvio condicional do Tipo-R que avalia se o conteúdo do registrador especificado é igual a zero antes de copiar o endereço alvo para o *Program Counter* (PC).
- (E) Os dois bits menos significativos do endereço alvo de qualquer desvio (condicional ou incondicional) são sempre descartados pelo hardware e preenchidos com 11 binário ao atualizar o PC, devido ao alinhamento de instruções de 32 bits.

Anotações



25

Considere o trecho de código MIPS a seguir.

```
funcao_principal:
# ... trecho inicial do código ...

li $s0, 10

addi $a0, $s0, -1

jal calcular_fatorial

add $s1, $v0, $s0

# ... restante do código ...

jr $ra
```

A função `funcao_principal` é uma função non-leaf chamada por outra função do programa e, portanto, deve retornar corretamente ao seu chamador ao final de sua execução. Além disso, a sub-rotina `calcular_fatorial` segue rigorosamente a convenção de chamadas do MIPS e realiza chamadas para outras sub-rotinas internas durante sua execução.

Considerando a convenção de chamadas do MIPS e o funcionamento da instrução `jal`, assinale a alternativa correta.

- (A) O código está correto, pois, como `$s0` é um registrador preservado pela função chamada (callee-saved), nenhum registrador utilizado em `funcao_principal` precisa ser salvo antes da chamada de `calcular_fatorial`.
- (B) A instrução `addi $a0, $s0, -1` é inválida, pois a ISA MIPS não permite imediatos negativos em operações aritméticas, e a correção consiste em substituir essa instrução por `subi $a0, $s0, 1`.
- (C) O erro ocorre porque a instrução `jal` não armazena automaticamente o endereço de retorno, consistindo a correção em substituir `jal calcular_fatorial` por `j calcular_fatorial`.

- (D) O registrador `$s0` será necessariamente destruído por `calcular_fatorial`; portanto, `funcao_principal` deve salvar `$s0` na pilha antes da chamada e restaurá-lo após o retorno.
- (E) O código contém um problema porque o valor original de retorno em `$ra` é sobrescrito, visto que a função `funcao_principal` também é uma sub-rotina chamada por outra função e realiza uma chamada adicional com `jal calcular_fatorial`; portanto, `funcao_principal` deve salvar o conteúdo de `$ra` (tipicamente na pilha) antes da chamada e restaurá-lo antes de executar `jr $ra`, garantindo o retorno correto ao seu chamador.

26

Considere a implementação de um processador MIPS com *pipeline* clássico de cinco estágios (IF, ID, EX, MEM e WB). Durante a execução de um programa, dependências de dados entre instruções podem introduzir penalidades de desempenho. Para mitigar esse problema, empregam-se mecanismos adicionais de hardware.

Nessa situação, a finalidade do mecanismo de encaminhamento (*forwarding* ou *bypassing*) é

- (A) duplicar unidades funcionais e barramentos internos, resolvendo conflitos estruturais decorrentes do uso simultâneo de recursos de hardware.
- (B) encaminhar resultados produzidos em estágios posteriores do *pipeline*, como EX/MEM ou MEM/WB, diretamente para unidades consumidoras desses dados, reduzindo a necessidade de interrupções (*stalls*) decorrentes de dependências de dados.
- (C) antecipar o fluxo de execução por meio de mecanismos de predição de desvios (*branch prediction*), reduzindo penalidades associadas a dependências de controle.
- (D) transferir resultados de operações aritméticas diretamente para o estágio de busca de instruções (IF), eliminando a necessidade de decodificação das instruções subsequentes.
- (E) eliminar completamente todos os *hazards* de dados do tipo RAW (*Read After Write*), inclusive aqueles decorrentes de instruções de carga imediatamente seguidas pelo uso do dado carregado (*load-use hazard*), dispensando qualquer inserção de *stalls*.

27

Considere a execução da instrução `lw $t0, 12($s1)` em uma implementação monociclo do processador MIPS.

Assinale a alternativa que descreve corretamente as operações realizadas pelo caminho de dados (*datapath*) durante a execução dessa instrução.

- (A) O conteúdo de `$s1` é somado ao deslocamento imediato após sua extensão de sinal; o endereço resultante é utilizado para leitura da memória de dados e o valor obtido é escrito em `$t0`.
- (B) O conteúdo de `$s1` é somado ao conteúdo de `$t0`; o resultado é enviado diretamente ao banco de registradores sem acesso à memória de dados.
- (C) O deslocamento imediato é enviado diretamente ao banco de registradores para selecionar `$t0`, enquanto a ULA compara esse valor com o conteúdo de `$s1`.
- (D) O conteúdo de `$t0` é combinado ao deslocamento imediato para gerar um endereço de escrita, ativando a memória de dados para armazenamento.
- (E) A memória de dados é acessada utilizando apenas o conteúdo de `$s1`; posteriormente, a ULA soma o deslocamento imediato ao valor lido antes da escrita em `$t0`.

28

No projeto de processadores baseados na arquitetura clássica MIPS, a microarquitetura da Unidade Central de Processamento (CPU) é estruturada a partir de dois blocos funcionais complementares: a Unidade de Controle (*Control Unit*) e o Caminho de Dados (*Datapath*).

Assinale a alternativa que descreve corretamente a relação e a coordenação operacional entre esses dois blocos durante a execução de uma instrução.

- (A) A unidade de controle realiza diretamente as operações aritméticas e lógicas da instrução, enquanto o caminho de dados limita-se à atualização do contador de programa (*Program Counter* – PC).
- (B) O caminho de dados interpreta os sinais gerados pela unidade de controle para executar as operações da instrução, sendo também responsável pela geração desses sinais a partir dos campos *Opcode* e *Funct*.

- (C) A unidade de controle decodifica os campos de especificação da instrução para gerar sinais de controle, como *RegWrite*, *ALUSrc* e *MemRead*, que determinam o comportamento dos componentes do processador, enquanto o caminho de dados provê os elementos responsáveis pelo processamento e pela transferência das informações, como banco de registradores, extensores de sinal, multiplexadores e ULA.
- (D) A unidade de controle e o caminho de dados desempenham funções redundantes, compartilhando os mesmos circuitos combinacionais e sequenciais para garantir tolerância a falhas durante a execução das instruções.
- (E) O caminho de dados interage exclusivamente com a memória de dados, enquanto a unidade de controle atua exclusivamente sobre o banco de registradores, sem interferir na seleção dos caminhos internos de informação.

29

Na arquitetura MIPS, diferentes instruções podem utilizar componentes comuns do caminho de dados (*datapath*), ainda que realizem operações distintas.

Considerando a relação existente entre a Arquitetura do Conjunto de Instruções (ISA) e a microarquitetura do processador, é correto afirmar que

- (A) instruções pertencentes a formatos distintos exigem necessariamente caminhos de dados independentes, impedindo o compartilhamento de recursos, como a ULA e o banco de registradores.
- (B) a separação entre ISA e microarquitetura elimina a necessidade de geração de sinais de controle específicos para cada instrução executada.
- (C) a microarquitetura determina o significado das instruções da ISA, alterando a semântica de operações aritméticas e de acesso à memória, sem comprometer a compatibilidade do software.
- (D) o compartilhamento de recursos entre instruções somente ocorre quando todas as instruções possuem exatamente o mesmo formato binário.
- (E) a existência de uma ISA bem definida permite que diferentes microarquiteturas implementem o mesmo conjunto de instruções, desde que preservem o comportamento observável especificado pela ISA.

30

Na arquitetura MIPS, as instruções possuem comprimento fixo de 32 bits e são organizadas em diferentes formatos, permitindo a codificação eficiente de operações aritméticas, acessos à memória e desvios.

Assinale a alternativa que descreve corretamente uma característica dos formatos de instrução utilizados nessa arquitetura.

- (A) O formato R possui um campo imediato de 16 bits, destinado ao armazenamento de constantes e endereços de memória.
- (B) O formato I contém campos para especificação de registradores e um campo imediato de 16 bits, incorporado à própria instrução.
- (C) O formato J utiliza campos destinados à identificação de três registradores de propósito geral e um campo de função (*Funct*).
- (D) Os formatos R e J possuem a mesma organização interna de campos, diferindo apenas no valor do campo opcode.
- (E) O formato I não possui campos destinados à especificação de registradores, utilizando apenas valores imediatos para determinar seus operandos.

31

As tecnologias de memórias voláteis e não voláteis possuem diferentes células de armazenamento.

Assinale a alternativa que caracteriza corretamente a diferença estrutural e física entre as células SRAM e DRAM.

- (A) SRAM usa capacitores dinâmicos e DRAM necessita de circuitos biestáveis.
- (B) SRAM exige ciclos periódicos de *refresh* e DRAM mantém os dados indefinidamente sob energia.
- (C) DRAM consome potência estática nula devido ao isolamento do silício.
- (D) A célula SRAM requer área física menor em comparação com DRAM.
- (E) SRAM armazena em inversores cruzados e DRAM em um capacitor.

32

A principal diferença entre um *latch* e um *flip-flop* é que

- (A) o *latch* é acionado por borda e o *flip-flop* por nível.
- (B) o *latch* armazena dois bits e o *flip-flop* apenas um.
- (C) o *latch* utiliza apenas transistores PMOS.
- (D) o *flip-flop* não armazena informação.
- (E) o *latch* é sensível ao nível do sinal de controle e o *flip-flop* é normalmente sensível à borda do sinal de relógio.

33

Considere um registrador de deslocamento para a direita de 4 bits, contendo inicialmente o valor 1011. Sendo a entrada serial igual a 0, qual será o conteúdo após um pulso de clock?

- (A) 0011
- (B) 0101
- (C) 0110
- (D) 1010
- (E) 1101

34

Em um circuito digital, a melhor definição de caminho crítico é o caminho entre dois elementos de armazenamento ou entrada e saída que apresenta

- (A) o maior atraso de propagação.
- (B) o menor número de portas lógicas.
- (C) o maior número de portas lógicas.
- (D) o menor atraso de propagação.
- (E) o maior número de conexões.

Anotações



35

O tempo de *setup* de um *flip-flop* corresponde ao intervalo

- (A) após a borda do clock, em que o dado deve permanecer estável.
- (B) antes da borda do clock, em que o dado deve permanecer estável.
- (C) necessário para propagação do clock.
- (D) entre dois pulsos consecutivos de clock.
- (E) necessário para inicialização do circuito.

36

Uma *Look-up Table* (LUT) de 4 entradas em um FPGA programado por SRAM implementa

- (A) apenas funções AND e OR de 4 entradas.
- (B) apenas funções booleanas com até 2 entradas.
- (C) qualquer função combinacional de até 4 variáveis.
- (D) apenas funções sequenciais.
- (E) qualquer função combinacional de até 6 variáveis.

37

O padrão IEEE define a estrutura para representação de números em ponto flutuante.

Os três campos constituintes de um número normalizado em precisão simples (32 bits) são

- (A) Sinal, Base Decimal e Mantissa.
- (B) Sinal, Expoente e Mantissa.
- (C) Magnitude, Mantissa e Bit de Sinal.
- (D) Complemento, Expoente Linear e Parte Fracionária.
- (E) Mantissa, Complemento e Expoente.

38

Sobre o consumo de potência em circuitos de lógica CMOS, é correto afirmar que o consumo dinâmico de potência ocorre principalmente devido à

- (A) corrente de fuga dos transistores.
- (B) resistência dos contatos metálicos.
- (C) carga e à descarga das capacitâncias do circuito durante a atividade de chaveamento.
- (D) polarização direta das junções PN.
- (E) temperatura ambiente.

39

Considere uma *Look-up Table* (LUT) com N entradas.

Quantos bits de memória são necessários para armazenar a programação da função booleana implementada por essa LUT em FPGA?

- (A) N
- (B) 2N
- (C) N^2
- (D) 2^N
- (E) N!

40

Um hazard estático ocorre em um circuito lógico combinacional quando

- (A) o sinal muda corretamente apenas uma vez.
- (B) o clock apresenta *jitter*.
- (C) o circuito entra em metastabilidade.
- (D) a saída deveria permanecer constante, mas apresenta uma transição temporária indesejada.
- (E) o consumo de potência aumenta.

Anotações



Anotações



CI INOVADOR

Instituições Executoras e Coexecutoras



UFSM



UnB



UFPR
UNIVERSIDADE FEDERAL DO PARANÁ



UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL



UNICAMP



UFC



UFCG
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE



unipampa
Universidade Federal do Pampa