Letra A

$$F = (\neg x \lor y) \land \neg(y \lor (x \land \neg z)) \lor (x \land \neg(y \land z))$$

Passo 1: Expandir os termos

Reescrevendo $\neg(y \land z)$ usando a **Lei de De Morgan**:

$$\neg(y \land z) = \neg y \lor \neg z$$

Substituímos na equação original:

$$F = (\neg x \lor y) \land \neg y \lor (x \land \neg z) \lor (x \land (\neg y \lor \neg z))$$

Aplicando a distributiva no último termo:

$$F = (\neg x \lor y) \land \neg y \lor (x \land \neg z) \lor (x \land \neg y) \lor (x \land \neg z)$$

Passo 2: Eliminar redundâncias

Observamos que $(x \land \neg z)$ aparece duas vezes, então podemos simplificá-lo:

$$F = (\neg x \lor y) \land \neg y \lor (x \land \neg y) \lor (x \land \neg z)$$

Agora, usando a distributiva na primeira parte:

$$(\neg x \lor y) \land \neg y = (\neg x \land \neg y) \lor (y \land \neg y)$$

Sabemos que $y \land \neg y = 0$, então sobra:

$$\neg x \land \neg y$$

Agora substituímos na equação geral:

$$F = (\neg x \land \neg y) \lor (x \land \neg y) \lor (x \land \neg z)$$

Passo 3: Fatoração

Reescrevendo os termos $(\neg x \land \neg y) \lor (x \land \neg y)$, podemos colocar $\neg y$ em evidência:

$$F = \neg y \land (\neg x \lor x) \lor (x \land \neg z)$$

Sabemos que $\neg x \lor x = 1$, então:

$$F = \neg y \lor (x \land \neg z)$$

Conclusão: Expressão Simplificada

A forma mais simples da expressão booleana é:

$$F = \neg y \lor (x \land \neg z)$$

Letra B

$$S = (\neg A \wedge \neg B \wedge Cin) \vee (\neg A \wedge B \wedge \neg Cin) \vee (A \wedge \neg B \wedge \neg Cin) \vee (A \wedge B \wedge Cin)$$

Passo 1: Construir a Tabela-Verdade

A expressão representa uma função booleana de três variáveis (A, B, Cin). Vamos montar a tabela verdade para identificar a simplificação.

A	B	Cin	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Os termos onde $S = 1$ correspondem às linhas:

- $\neg A \wedge \neg B \wedge Cin$
- $\neg A \wedge B \wedge \neg Cin$
- $A \wedge \neg B \wedge \neg Cin$
- $A \wedge B \wedge Cin$

Passo 2: Identificar o padrão

Observando os termos, percebemos que esta é a **função soma** (S) de um meio somador (half-adder):

$$S = A \oplus B \oplus Cin$$

Conclusão

A forma mais simples da expressão é:

$$S = A \oplus B \oplus Cin$$

Letra C

$RegDst = \neg S2 \wedge S1 \wedge \neg S0 \wedge \neg Op5 \wedge \neg Op4 \wedge \neg Op3 \wedge \neg Op2 \wedge \neg Op1 \wedge \neg Op0$

Passo 1: Análise da Expressão

Essa expressão representa um caso específico em que:

- $S2 = 0$
- $S1 = 1$
- $S0 = 0$
- $Op5, Op4, Op3, Op2, Op1, Op0 = 0$ (Ou seja, todos os bits de Op são 0)

Isso significa que $RegDst$ só será ativado quando os sinais de controle S e Op assumirem esses valores específicos.

Passo 2: Identificação da Simplificação

- A parte $\neg Op5 \wedge \neg Op4 \wedge \neg Op3 \wedge \neg Op2 \wedge \neg Op1 \wedge \neg Op0$ pode ser escrita como $Op = 000000$, que geralmente representa o tipo R no conjunto de instruções MIPS.
- A parte $\neg S2 \wedge S1 \wedge \neg S0$ indica uma configuração específica dos sinais S.

Conclusão

A expressão já está na sua forma mais simplificada em termos booleanos, mas pode ser reescrita de forma mais intuitiva como:

$$RegDst = (S = 010) \wedge (Op = 000000)$$

Ou, se for no contexto de MIPS:

$$RegDst = (S = 010) \wedge \text{Instrução Tipo R}$$

Letra D

$F = (ABCD) \vee (AB\neg CD) \vee (AB\neg C\neg D) \vee (A\neg BCD) \vee (\neg ABCD) \vee (\neg ABC\neg D) \vee (\neg AB\neg CD) \vee (\neg A\neg B\neg CD)$

1. **Identificar Termos Comuns**: Procure por termos que podem ser combinados usando a propriedade distributiva.
2. **Aplicar as Regras da Álgebra Booleana**:
   - $A \vee \overline{A} = 1$
   - $A \wedge 1 = A$
   - $A \wedge \overline{A} = 0$
3. **Simplificar a Expressão**:
   - Combine termos que diferem por apenas uma variável.

Após a simplificação, a expressão se reduz a:

$$F = AB \vee A\overline{B} \vee \overline{A}B \vee \overline{AB}$$

Isso pode ser simplificado ainda mais para:

$$F = 1$$

Isso significa que a função F é sempre verdadeira, independentemente dos valores de entrada de $A, B, C,$ e D. Esse resultado sugere que a expressão original cobre todas as combinações possíveis das variáveis, tornando F uma tautologia.

Letra E

$$XOR : S_i = A_i \oplus B_i \quad NAND : S_i = \neg(A_i \wedge B_i) \quad NOR : S_i = \neg(A_i \vee B_i)$$

1. **XOR (Ou Exclusivo)**:

$$S_i = A_i \oplus B_i$$

A operação XOR pode ser expressa como:

$$S_i = (A_i \wedge \overline{B_i}) \vee (\overline{A_i} \wedge B_i)$$

Essa é a forma mais simples e direta para a operação XOR.

2. **NAND (Não E)**:

$$S_i = \overline{(A_i \wedge B_i)}$$

Aplicando a lei de De Morgan:

$$S_i = \overline{A_i} \vee \overline{B_i}$$

Essa é a forma mais simplificada para a operação NAND.

3. **NOR (Não Ou)**:

$$S_i = \overline{(A_i \vee B_i)}$$

Aplicando a lei de De Morgan:

$$S_i = \overline{A_i} \wedge \overline{B_i}$$

Essa é a forma mais simplificada para a operação NOR.

Essas são as formas definitivamente simplificadas das operações lógicas XOR, NAND e NOR. Cada uma delas está na sua forma mais reduzida, utilizando as operações básicas da álgebra booleana.

Letra F

$$F=(pc\_update \wedge mem\_read\_instr) \wedge (instr\_decode \wedge reg\_read\_1 \wedge reg\_read\_2) \wedge (alu\_control \wedge alu\_execute) \wedge reg\_write$$

Podemos utilizar a propriedade associativa da operação AND (
$\wedge$
$\wedge$), que nos permite agrupar os termos de maneira conveniente. A propriedade associativa diz que:

$$(\mathrm{A} \wedge \mathrm{B}) \wedge \mathrm{C}=\mathrm{A} \wedge (\mathrm{B} \wedge \mathrm{C})$$

$$(A \wedge B) \wedge C=A \wedge (B \wedge C)$$

Aplicando essa propriedade, podemos reescrever a expressão como:

$$\mathrm{F=pc\_update \wedge mem\_read\_instr \wedge instr\_decode \wedge reg\_read\_1 \wedge reg\_read\_2 \wedge alu\_control \wedge alu\_execute \wedge reg\_write}$$

$$F=pc\_update \wedge mem\_read\_instr \wedge instr\_decode \wedge reg\_read\_1 \wedge reg\_read\_2 \wedge alu\_control \wedge alu\_execute \wedge reg\_write$$

Essa é a forma mais simplificada da expressão, onde todos os termos estão conectados por operações AND. Não há simplificação adicional possível sem mais informações sobre as relações entre as variáveis ou possíveis redundâncias. Portanto, a expressão final simplificada é:

$$\mathrm{F=pc\_update \wedge mem\_read\_instr \wedge instr\_decode \wedge reg\_read\_1 \wedge reg\_read\_2 \wedge alu\_control \wedge alu\_execute \wedge reg\_write}$$

$$F=pc\_update \wedge mem\_read\_instr \wedge instr\_decode \wedge reg\_read\_1 \wedge reg\_read\_2 \wedge alu\_control \wedge$$