Taller de Lógica Digital

Organización del Computador 1

Primer Cuatrimestre 2021

3. Antes de Empezar

Completar la siguiente tabla indicando los resultados para Op1 + Op2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operandos** | | **Sin Signo** | | | | | **Complemento a 2** | | | | |
| **Op1** | **Op2** | **Op110** | **Op210** | **Res(bits)** | **Res10** | **V?** | **Op110** | **Op210** | **Res(bits)** | **Res10** | **V?** |
| **1111** | **0001** | 15 | 1 | 0000 | 0 | 1 | -1 | 1 | 0000 | 0 | 0 |
| **0001** | **1111** | 1 | 15 | 0000 | 0 | 1 | 1 | -1 | 0000 | 0 | 0 |
| **0101** | **0101** | 5 | 5 | 1010 | 10 | 0 | 5 | 5 | 1010 | -6 | 1 |
| **1000** | **0111** | 8 | 7 | 1111 | 15 | 0 | -8 | 7 | 1111 | -1 | 0 |
| **0110** | **1010** | 6 | 10 | 0000 | 0 | 1 | 6 | -6 | 0000 | 0 | 0 |

Completar la siguiente tabla indicando los resultados para Op1 - Op2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operandos** | | **Sin Signo** | | | | | **Complemento a 2** | | | | |
| **Op1** | **Op2** | **Op110** | **Op210** | **Res(bits)** | **Res10** | **V?** | **Op110** | **Op210** | **Res(bits)** | **Res10** | **V?** |
| **1000** | **0010** | 8 | 2 | 0110 | 6 | 0 | -8 | 2 | 0110 | 6 | 1 |
| **0001** | **1111** | 1 | 15 | 0010 | 2 | 1 | 1 | -1 | 0010 | 2 | 0 |
| **0101** | **0101** | 5 | 5 | 0000 | 0 | 0 | 5 | 5 | 0000 | 0 | 0 |
| **1000** | **0111** | 8 | 7 | 0001 | 1 | 0 | -8 | 8 | 0001 | 1 | 1 |
| **0110** | **1010** | 6 | 10 | 1100 | 12 | 1 | 6 | -6 | 1100 | -4 | 1 |

5. Validación de los resultados

Completar la siguiente tabla indicando los resultados utilizando la ALU de 4 bits.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operandos** | | **Sumador** | | | | | **Restador** | | | | |
| **A** | **B** | **S** | **Z** | **C** | **V** | **N** | **S** | **Z** | **C** | **V** | **N** |
| **1111** | **0001** | 0000 | 1 | 1 | 0 | 0 | 1110 | 0 | 0 | 0 | 1 |
| **0001** | **1111** | 0000 |  |  |  |  |  |  |  |  |  |
| **0101** | **0101** | 1010 | 0 | 0 | 1 | 1 | 0001 | 0 | 0 | 0 | 0 |
| **1000** | **0111** | 1111 | 0 | 0 | 0 | 1 | 0001 | 0 | 0 | 0 | 0 |
| **0110** | **1010** | 0000 | 1 | 1 | 0 | 0 | 1100 | 0 | 1 | 0 | 1 |

i) ¿Es posible utilizar esta ALU con operandos interpretados como sin signo?¿Cómo de-

tectarı́a el overflow

Si se suman operandos sin signo el carry sería la forma de detectar un overflow.

Ej: 1111 + 0001 = 1 0000

En este caso requerimos de un quinto dígito para representar correctamente el número, lo que sería en esencia un overflow.