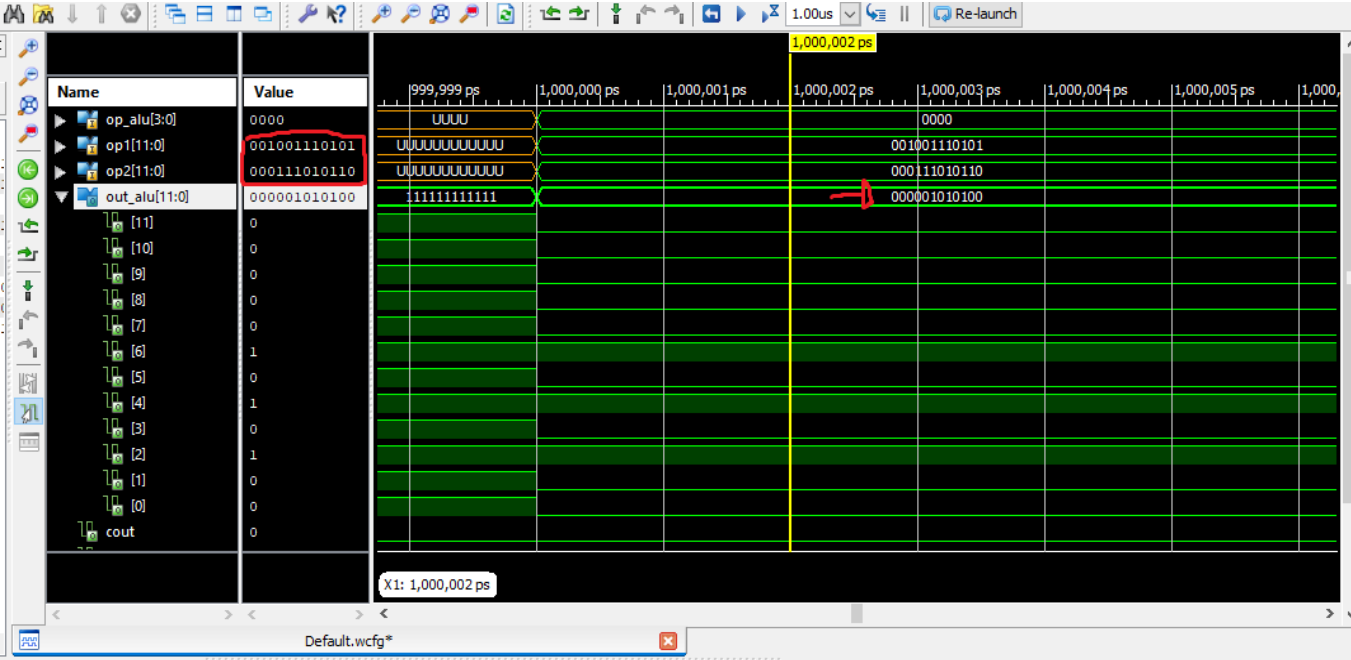
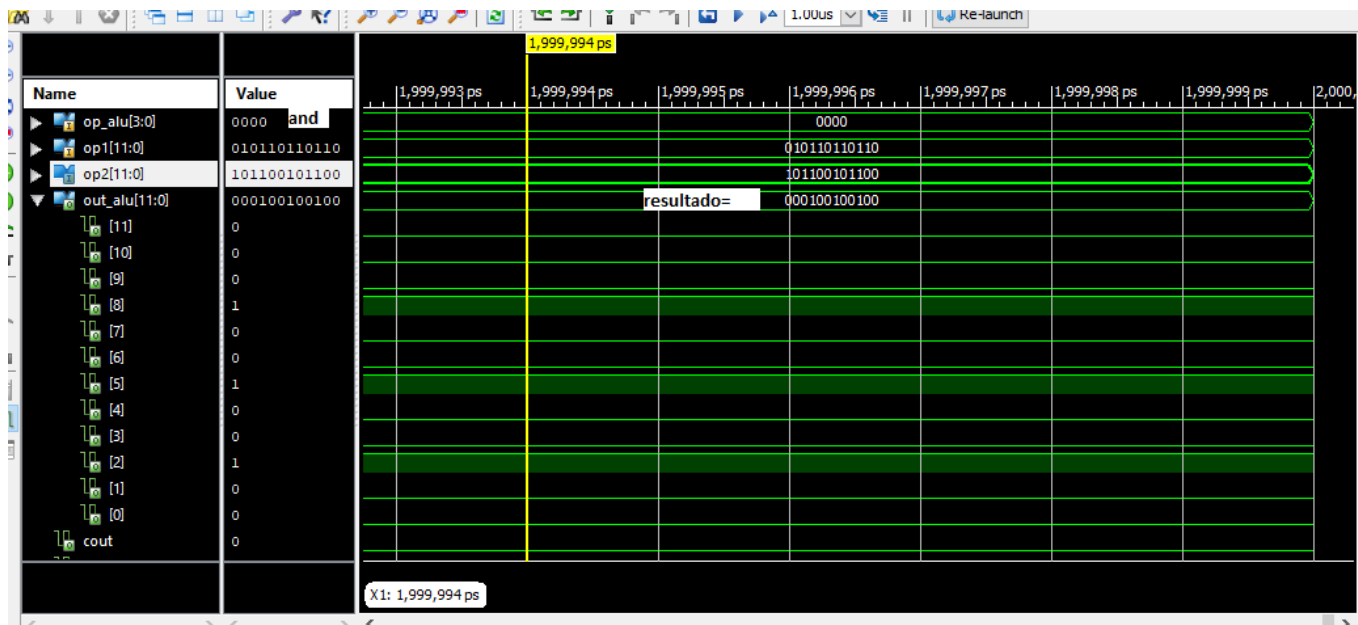


Aluno: Igor Pereira Dourado 19204004

operação	Caso 1	Caso 2
AND	<div>001001110101</div> <div>AND 000111010110</div> <div>-----</div> <div>000001010100</div> <div> VALOR EM HEXA: 54</div>	<div>010110110110</div> <div>AND 101100101100</div> <div>-----</div> <div>000100100100</div> <div> VALOR EM HEXA: 124</div>

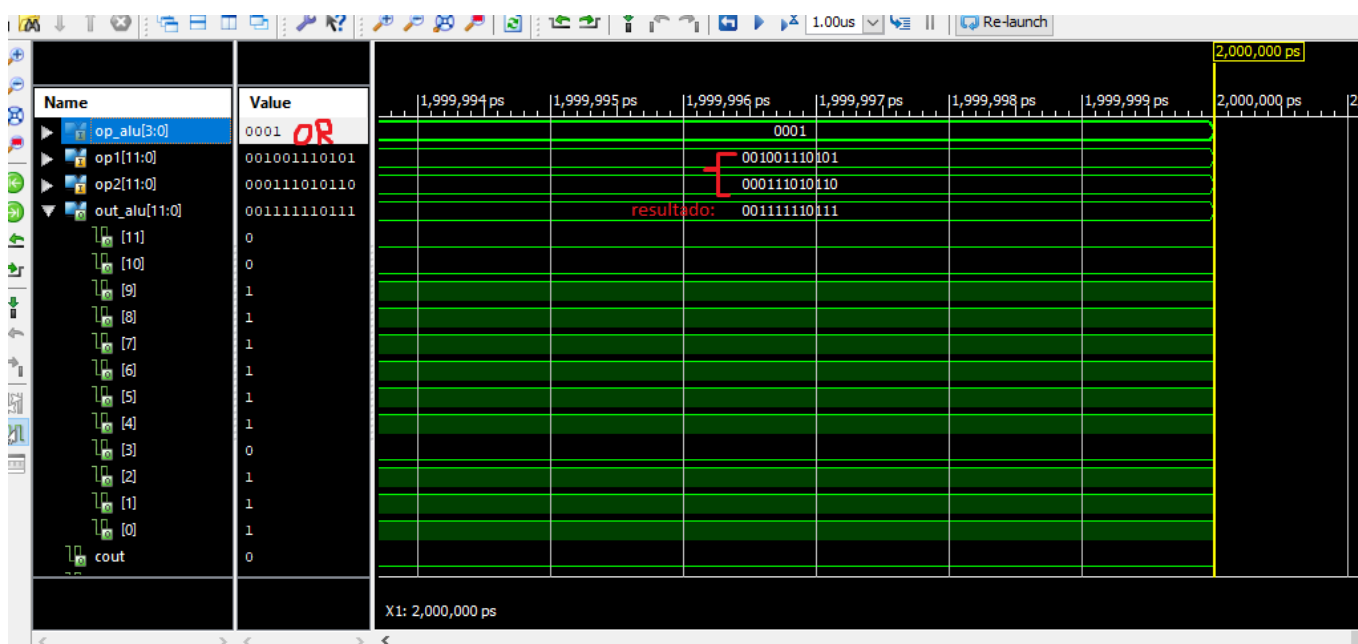


Caso 1 de And

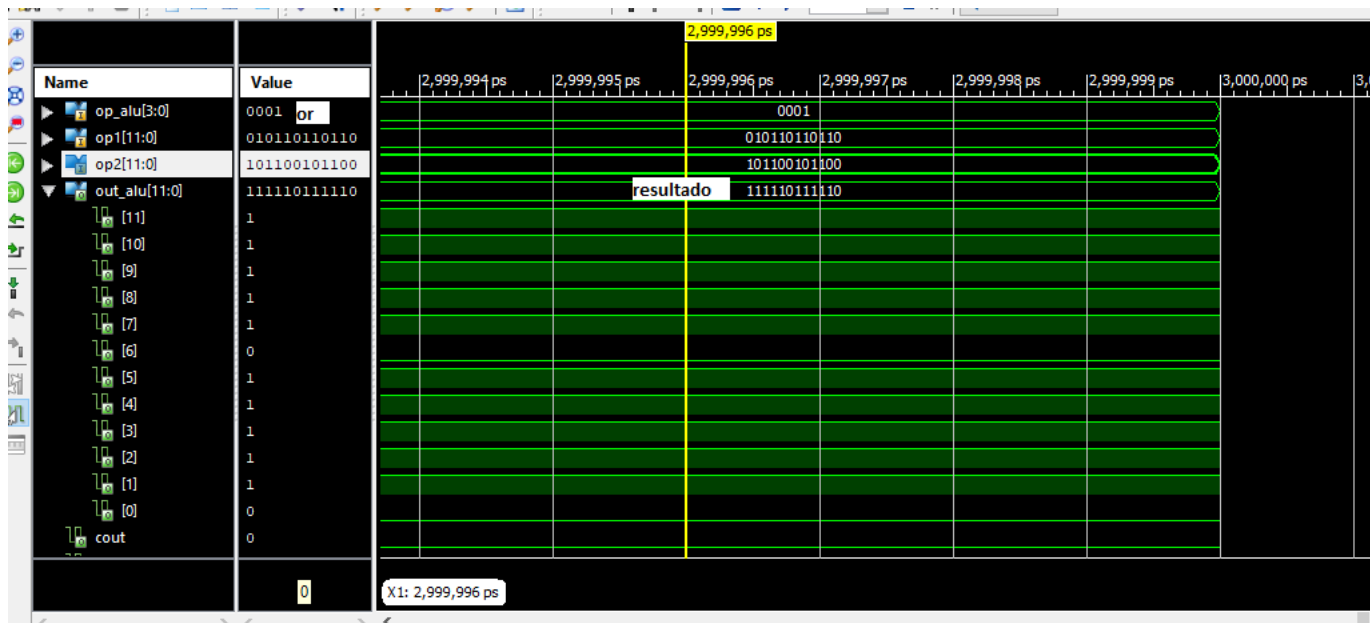


Caso 2 de And

operação	Caso 1	Caso 2
OR	<pre> 001001110101 000111010110 ----- 001111110111 </pre> <p>VALOR EM HEXA: 3F7</p>	<pre> 010110110110 101100101100 ----- 111110111110 </pre> <p>VALOR EM HEXA: FBE</p>

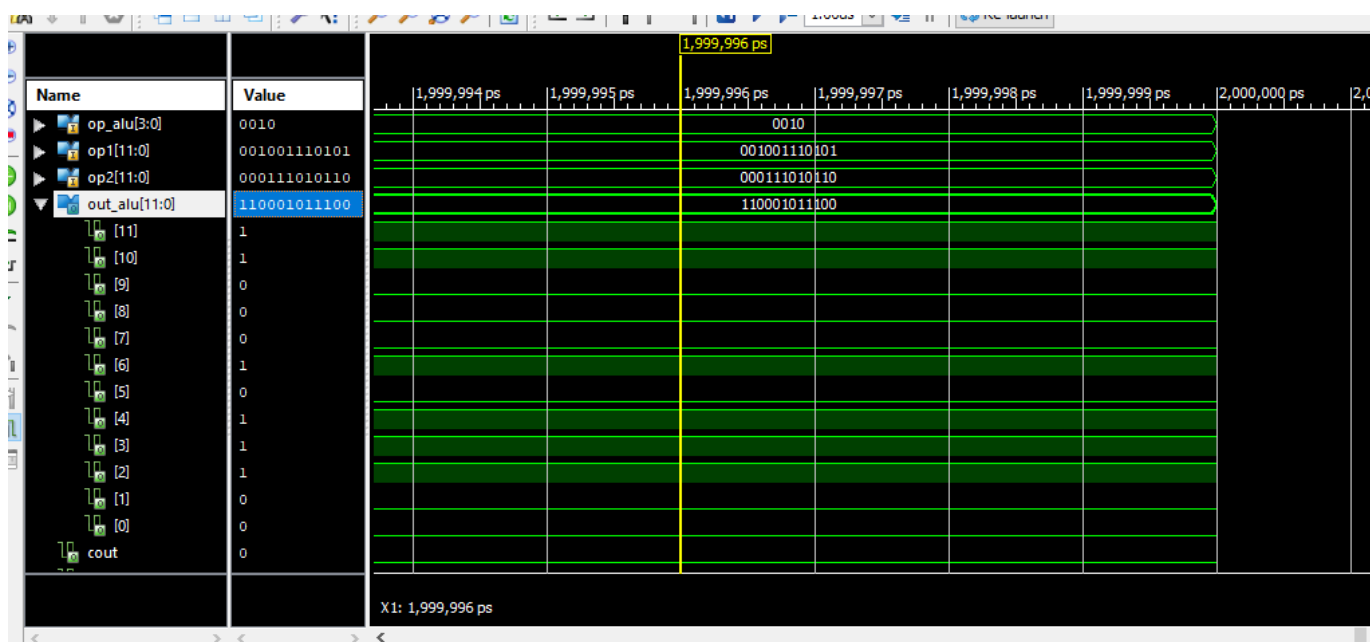


Caso 1 de Or

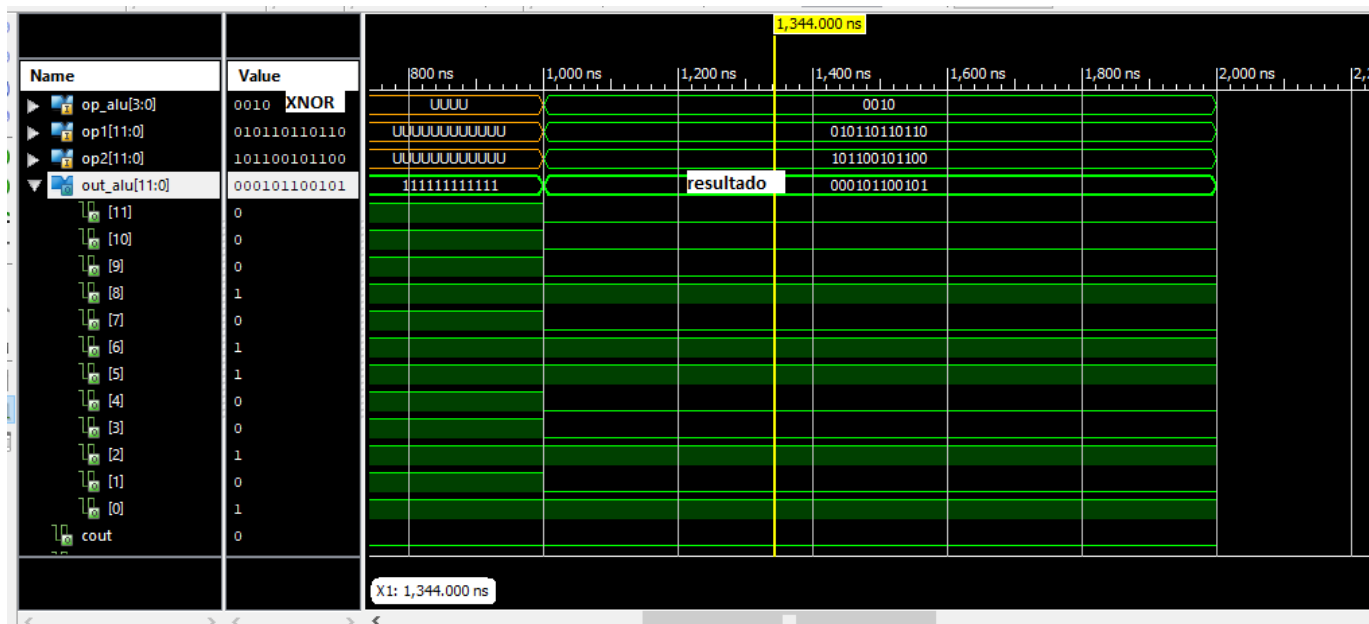


Caso 2 de Or

operação	Caso 1	Caso 2
XNOR	001001110101	010110110110
	XNOR 000111010110	XNOR 101100101100
	-----	-----
	110001011100	000101100101
	VALOR EM HEXA: C5C	VALOR EM HEXA: 165

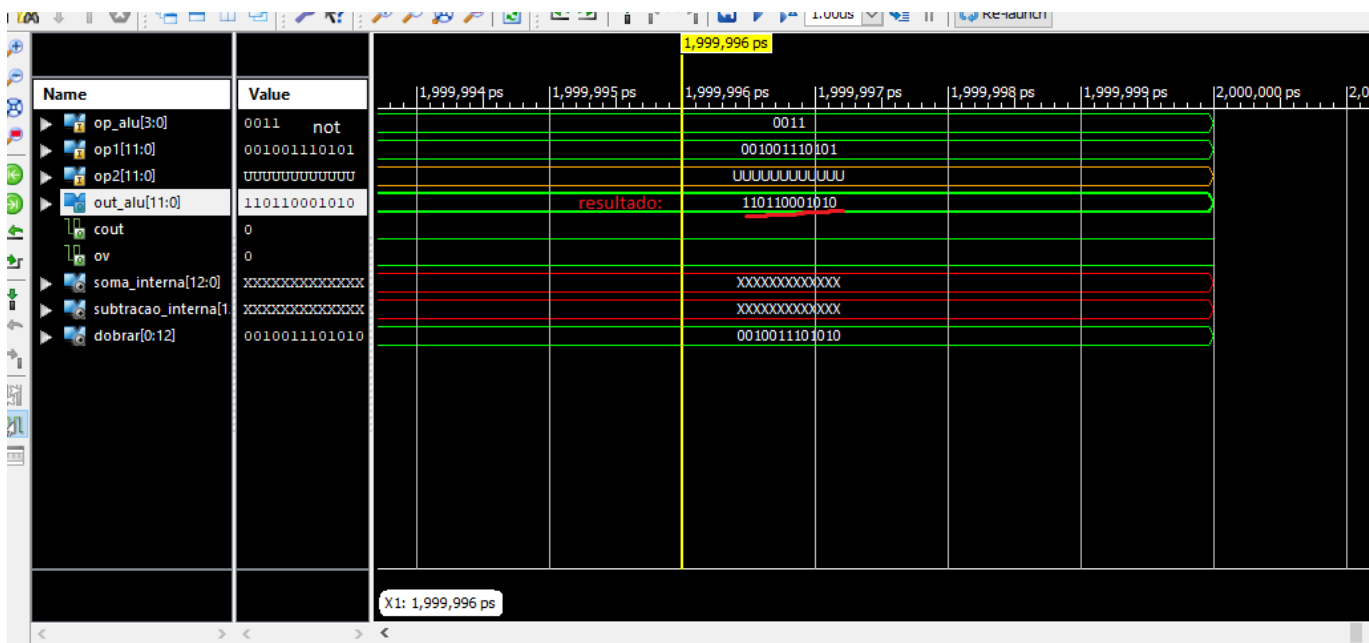


Caso 1 de Xnor

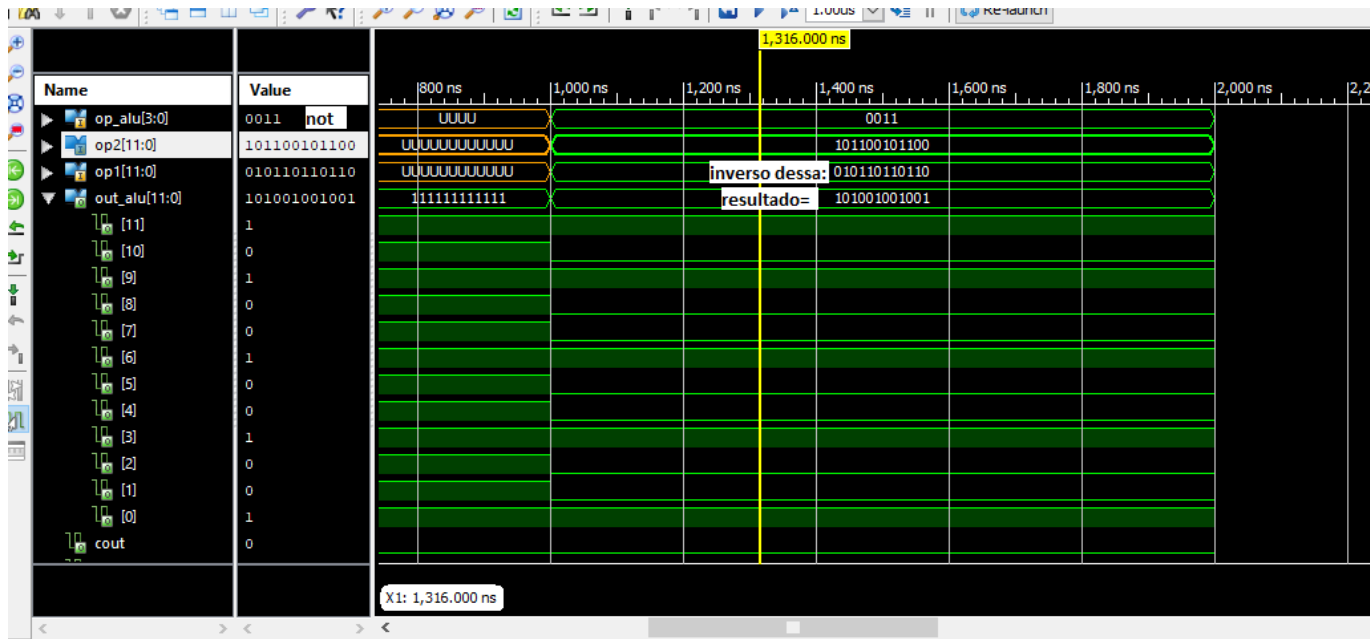


Caso 2 de Xnor

operação	Caso 1	Caso 2
NOT	<pre> 001001110101 ----- 110110001010 </pre> <p>VALOR EM HEXA: D8A</p>	<pre> 010110110110 ----- 101001001001 </pre> <p>VALOR EM HEXA: A49</p>

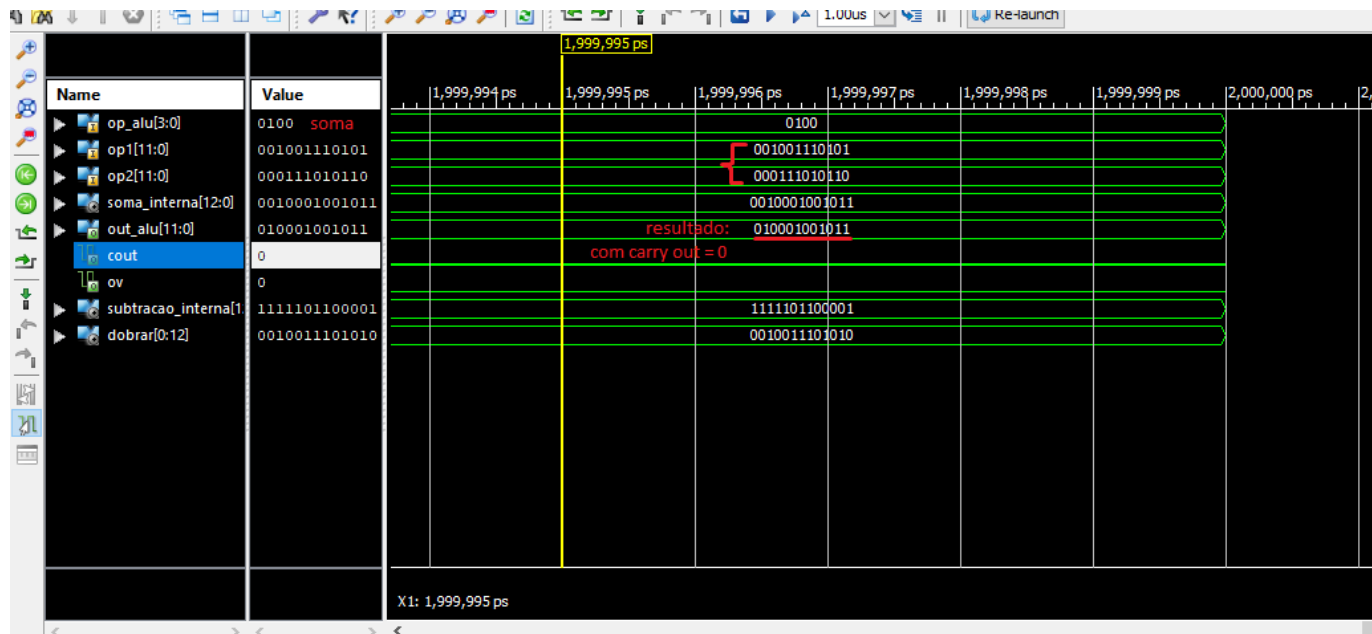


Caso 1 de Not

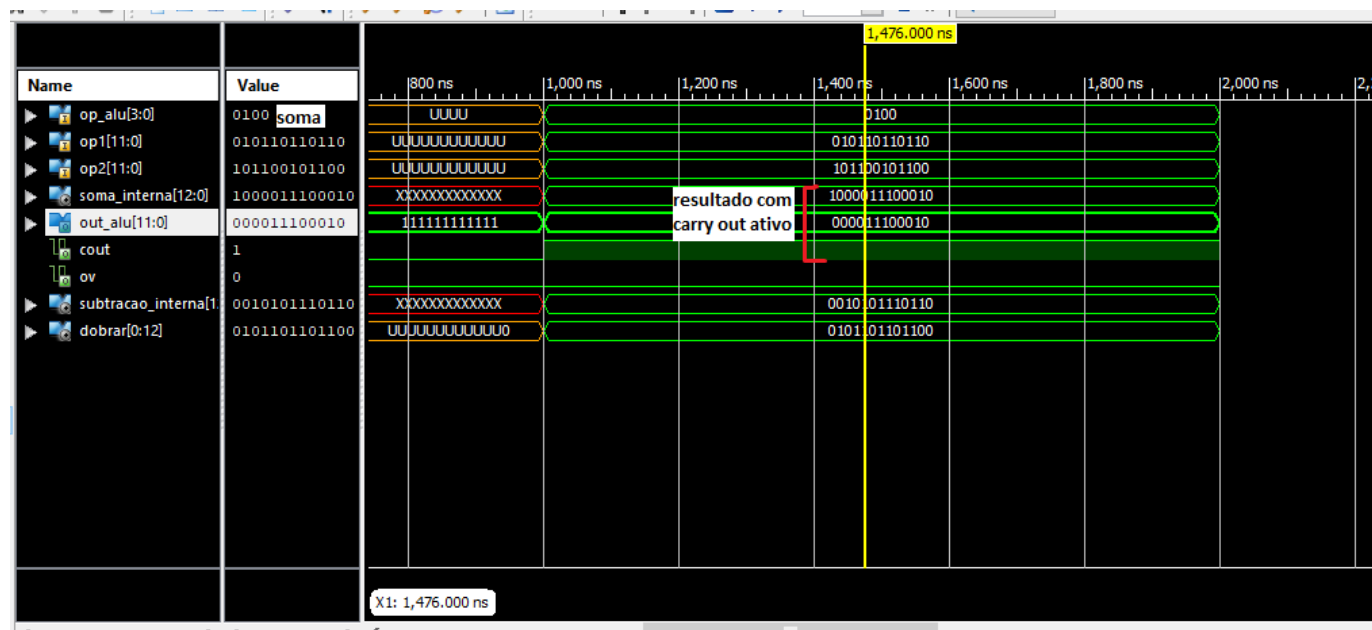


Caso 2 de Not

operação	Caso 1	Caso 2
SOMA	<pre> 001001110101 + 000111010110 ----- 010001001011 </pre> <p>CARRY OUT= 0 VALOR EM HEXA: 44B *O carry out ocorre caso haja um excesso na soma dos bits em questão, e o carry out representa esse valor que vai passar para ser somado com os bits da próxima fileira de soma</p>	<pre> 010110110110 + 101100101100 ----- 1 000011100010 </pre> <p>CARRY OUT = 1 VALOR EM HEXA: 10E2 *Nesse caso o carry out final recebe 1, visto que devido a soma da ultima fileira de bits, passará "1" para a próxima fileira, mas como devido ao tamanho do vetor ela não existe, o carry out vai como primeiro bit da representação</p>

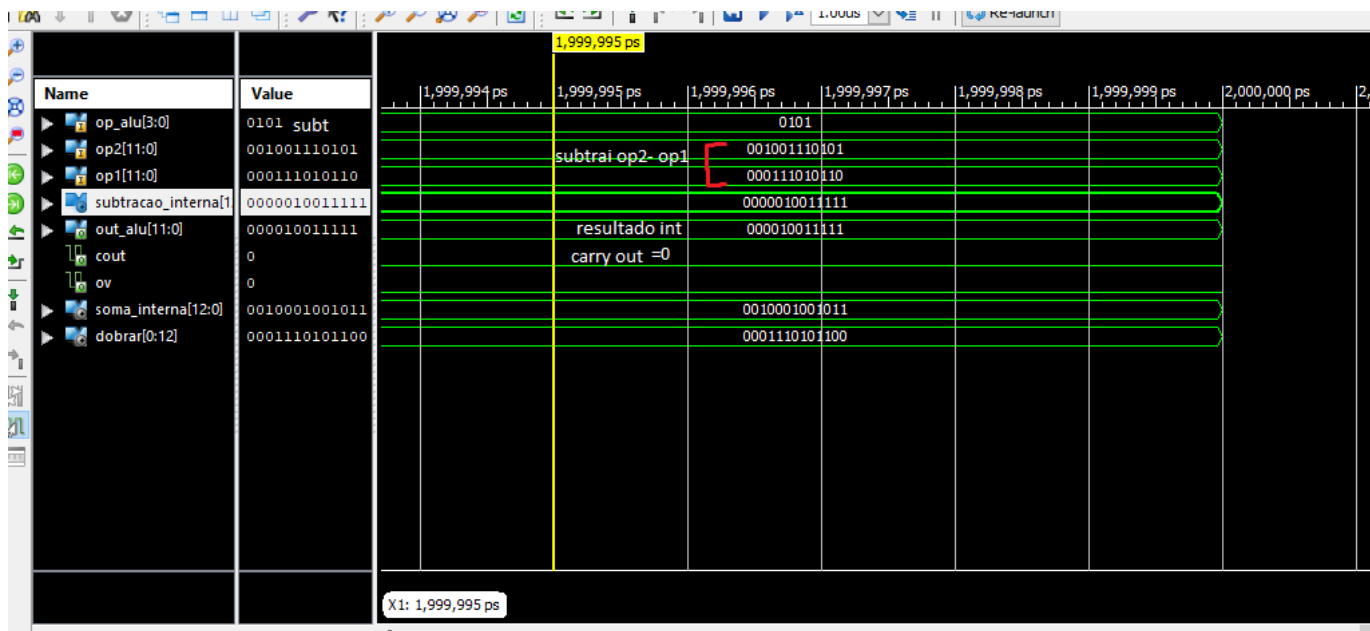


Caso 1 de Soma

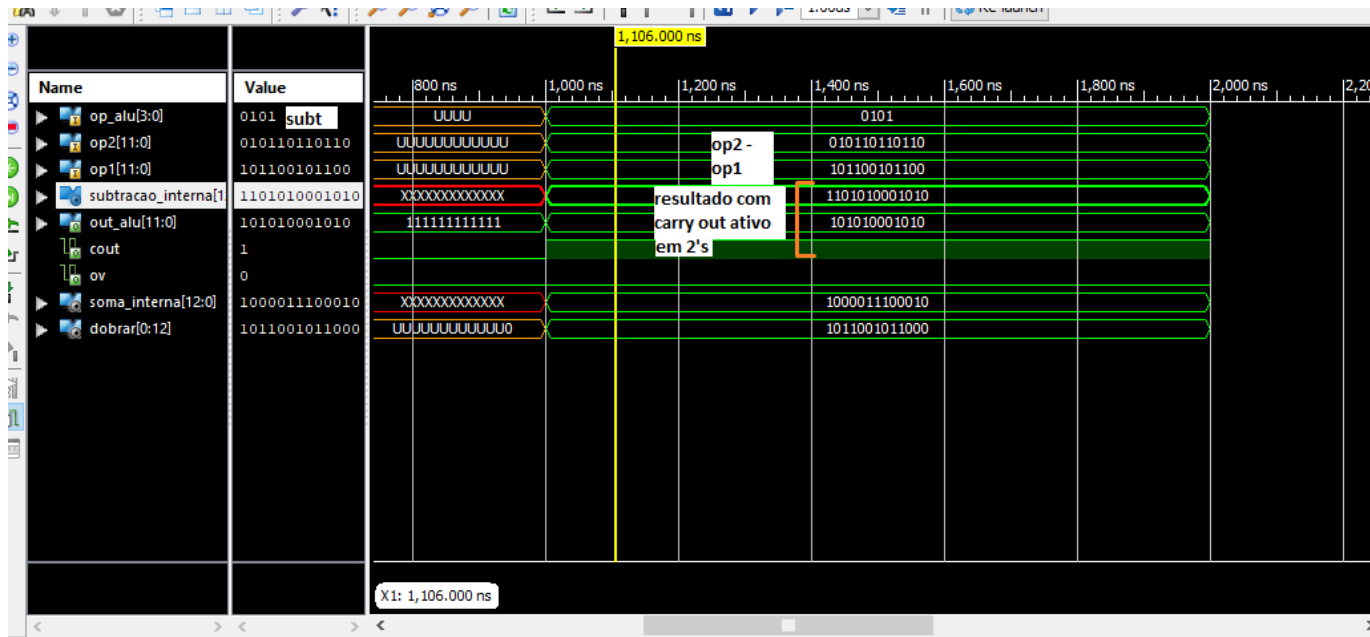


Caso 2 de Soma

operação	Caso 1	Caso 2
SUBTRAÇÃO	<pre> 001001110101 - 000111010110 ----- 0 000010011111 VALOR EM HEXA: 9F COUT=0 *O carry out aqui ocorre caso a subtração dos bits em questão precise “pegar emprestado” 1 bit do próximo, e o carry out representa esse valor que vai passar para ser subtraído com os bits da próxima fileira de subtração </pre>	<pre> 010110110110 - 101100101100 ----- 1 101010001010 (resultado em complement de dois) CARRY OUT = 1 VALOR EM HEXA: -576 *O carry out recebe 1 pois o ultimo bit precisaria “pegar emprestado” 1 bit da próxima fileira, porém essa fileira não é representada devido ao tamanho do vetor </pre>

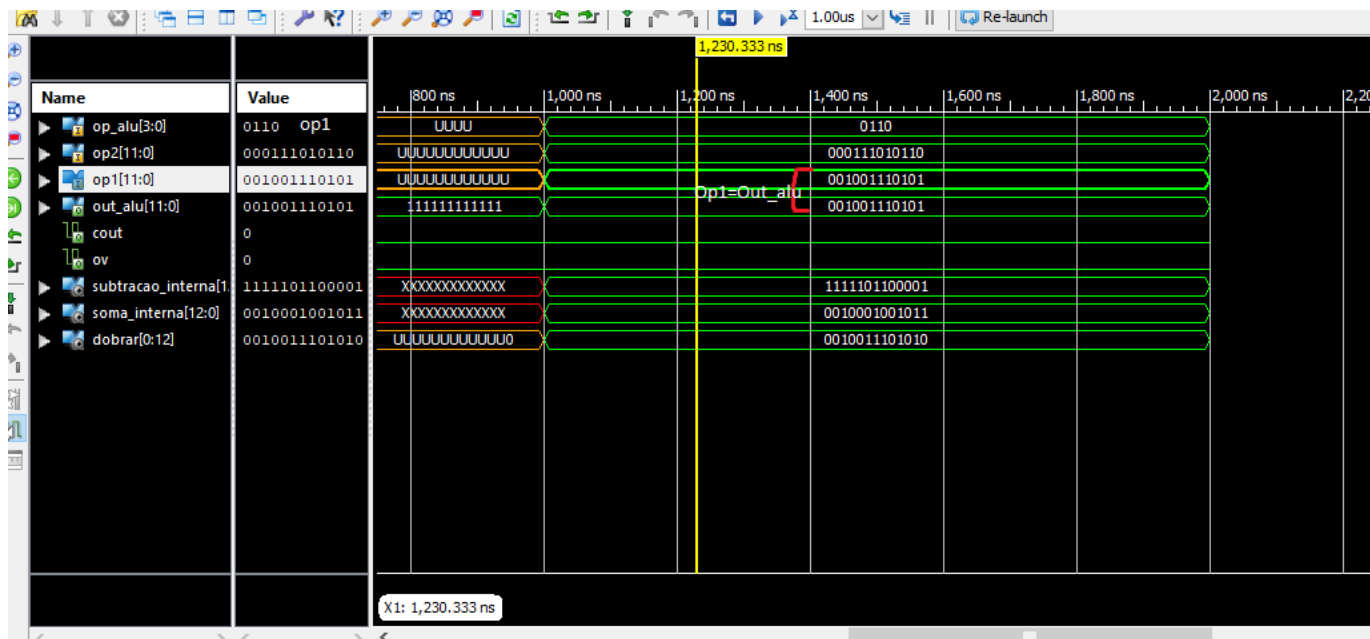


Caso 1 de Subtração

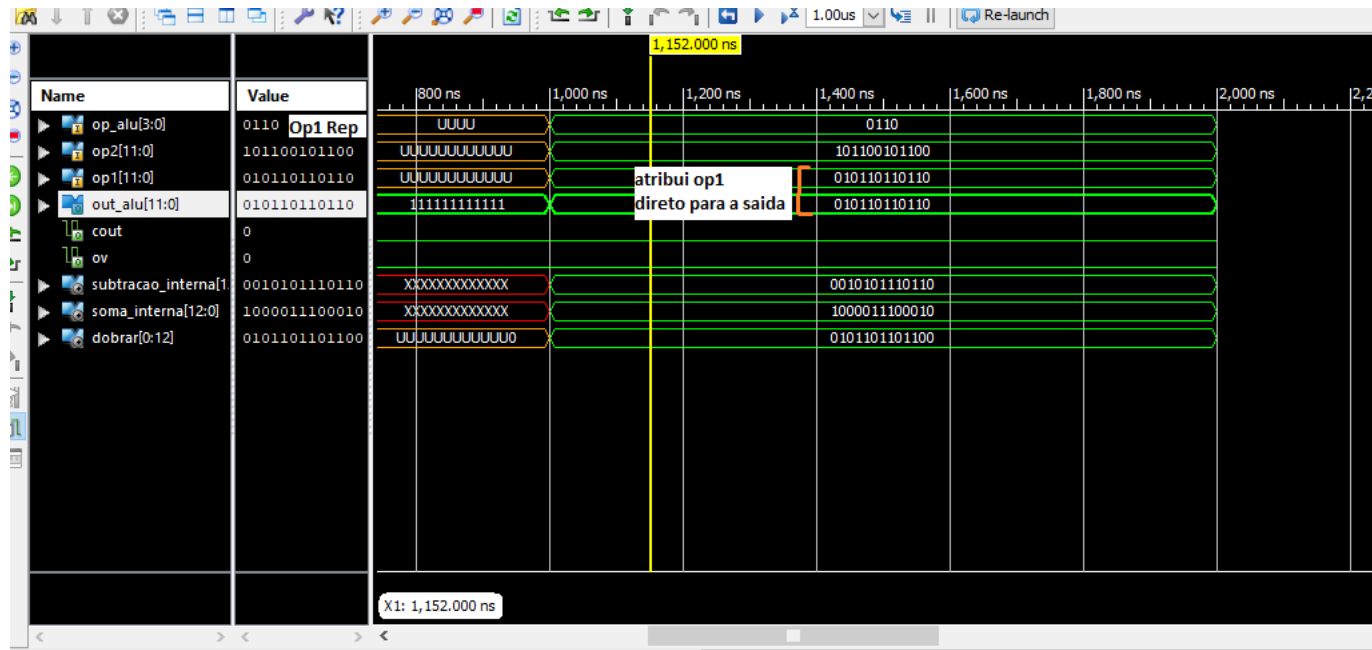


Caso 2 de Subtração

operação	Caso 1	Caso 2
OP1	<pre> 001001110101 ----- 001001110101 </pre> <p>VALOR EM HEXA: 275</p>	<pre> 010110110110 ----- 010110110110 </pre> <p>VALOR EM HEXA: 5B6</p>



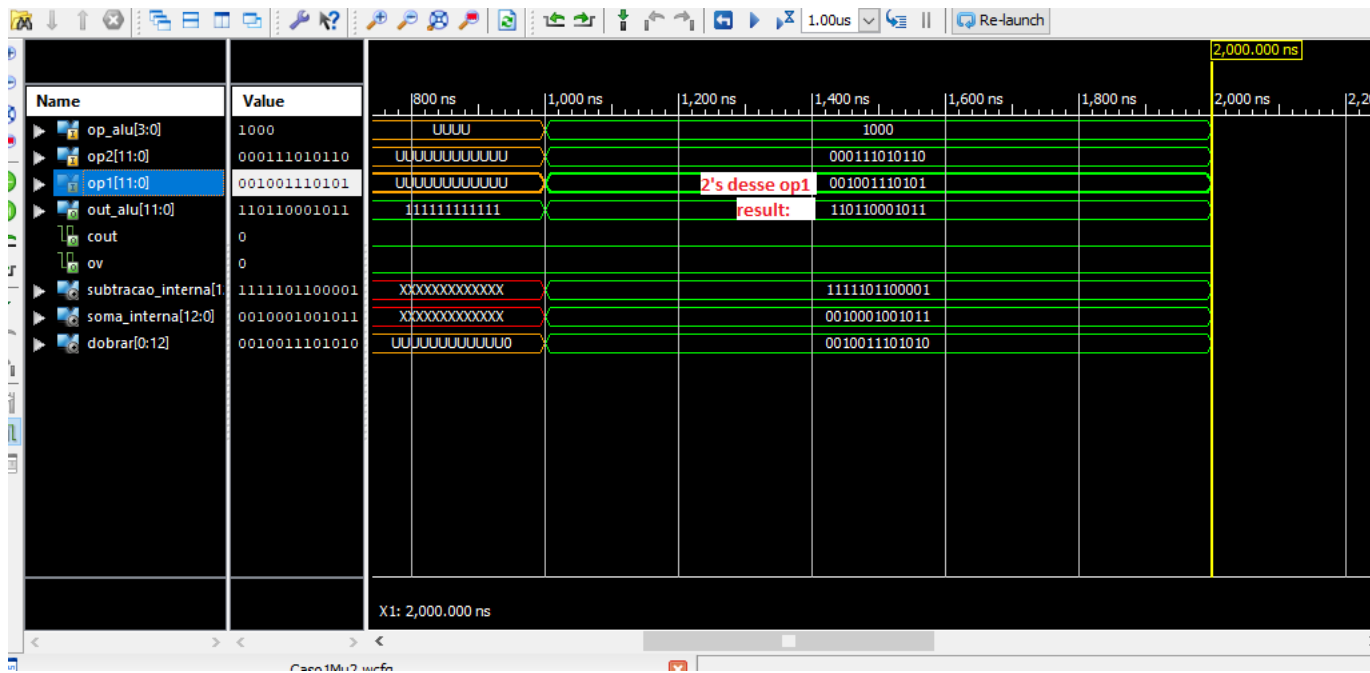
Caso 1 de Op1



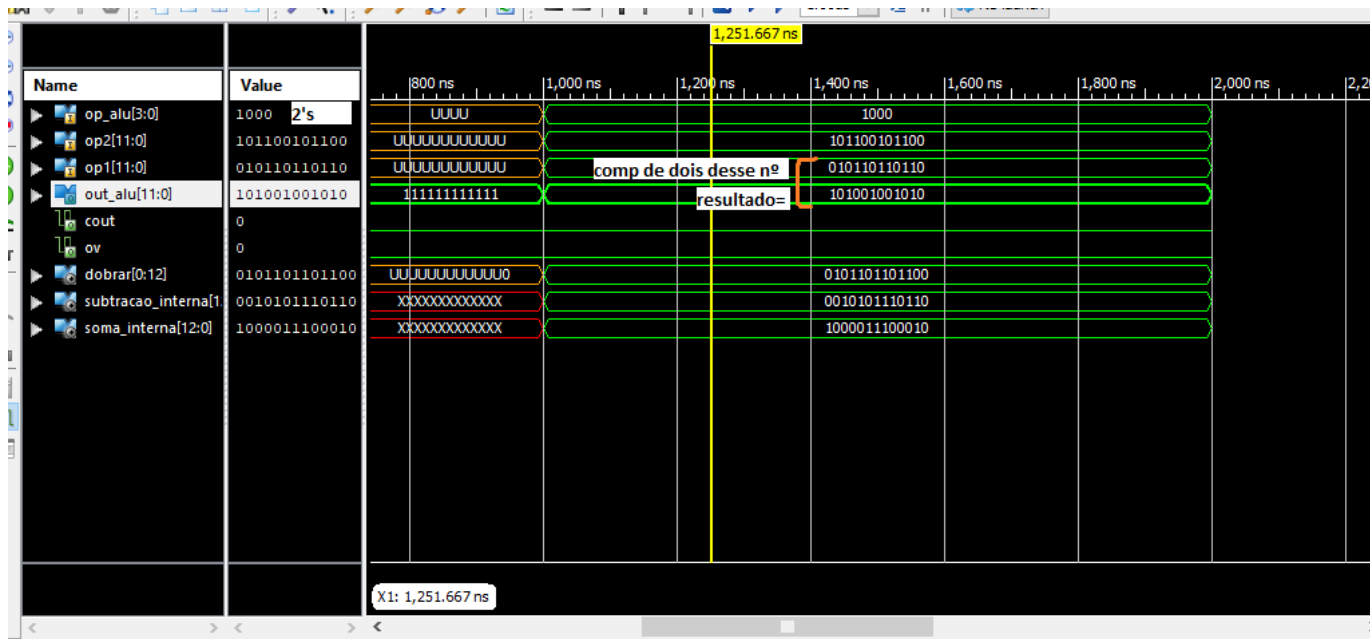
Caso 2 de Op1

operação	Caso 1	Caso 2
MU2	$ \begin{array}{r} 001001110101 \\ + \quad 001001110101 \\ \hline 010011101010 \end{array} $ <p>OVERFLOW= 0, não ocorre VALOR EM HEXA: 4EA</p>	$ \begin{array}{r} 101100101100 \\ + \quad 101100101100 \\ \hline \underline{1} \quad 011001011000 \end{array} $ <p>OVERFLOW= 1, ocorre pois o valor da multiplicação por 2 não é possível de se representar com o vetor dispondo apenas do total de bits que ele dispõe, ocorrendo overflow VALOR EM HEXA: 1658</p>

operação	Caso 1	Caso 2
2's	<p>001001110101 (inverte abaixo)</p> <p>110110001010</p> <p>+ 1</p> <p>-----</p> <p>110110001011</p> <p>VALOR EM HEXA: -D8B</p>	<p>010110110110 (inverte abaixo)</p> <p>101001001001</p> <p>+ 1</p> <p>-----</p> <p>101001001010</p> <p>VALOR EM HEXA: -5B7</p>



Caso 1 de Complemento de dois



Caso 2 do Complemento de dois