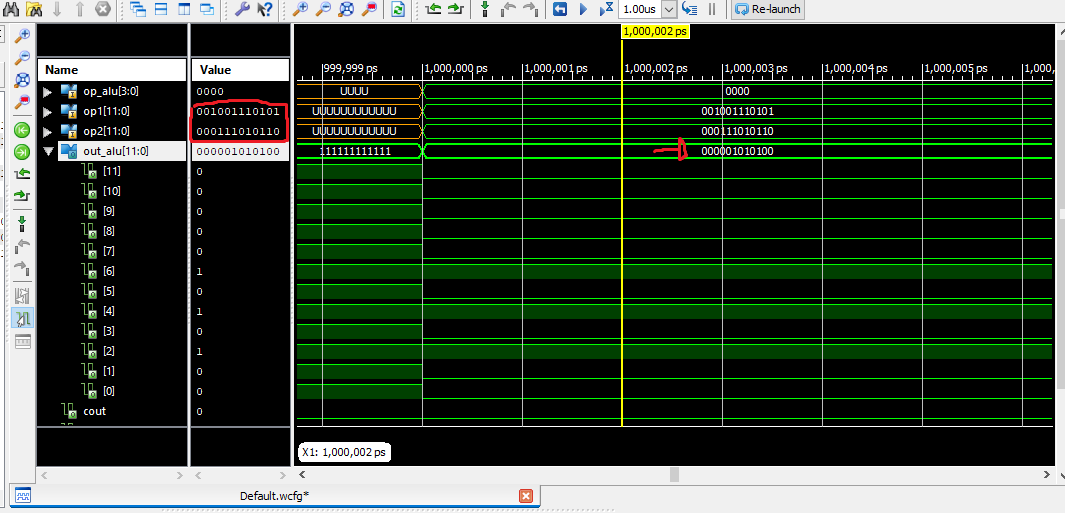
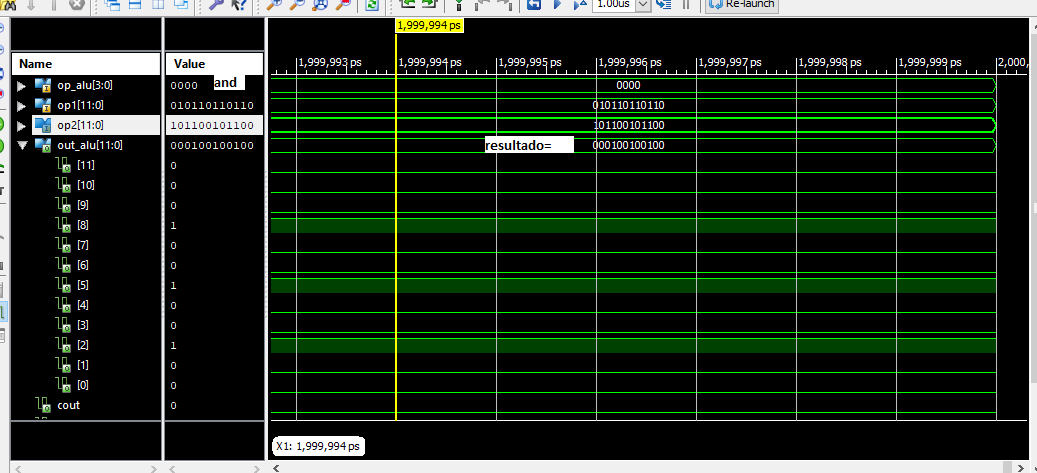
Aluno: Igor Pereira Dourado 19204004

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| AND | 001001110101  AND 000111010110  --------------------  000001010100  VALOR EM HEXA: 54 | 010110110110  AND 101100101100  --------------------  000100100100  VALOR EM HEXA: 124 |

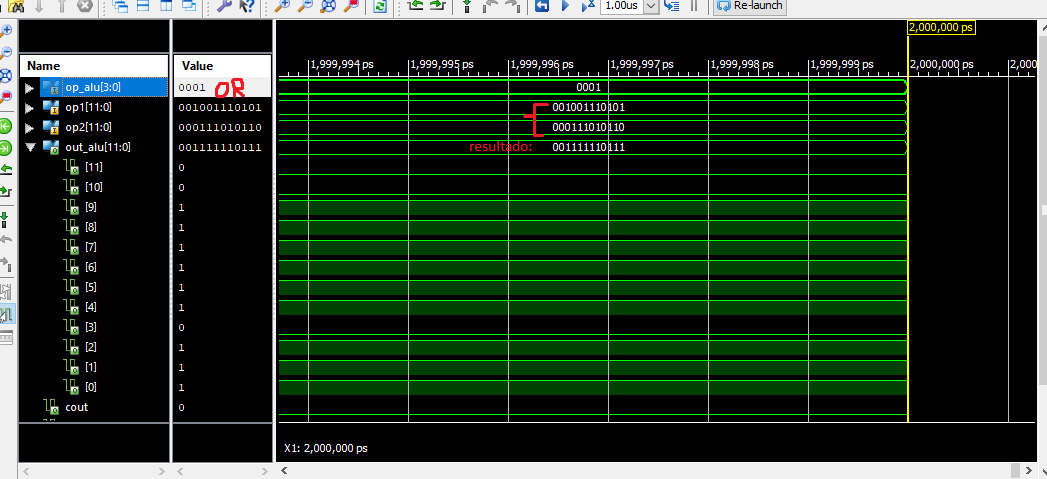


Caso 1 de And

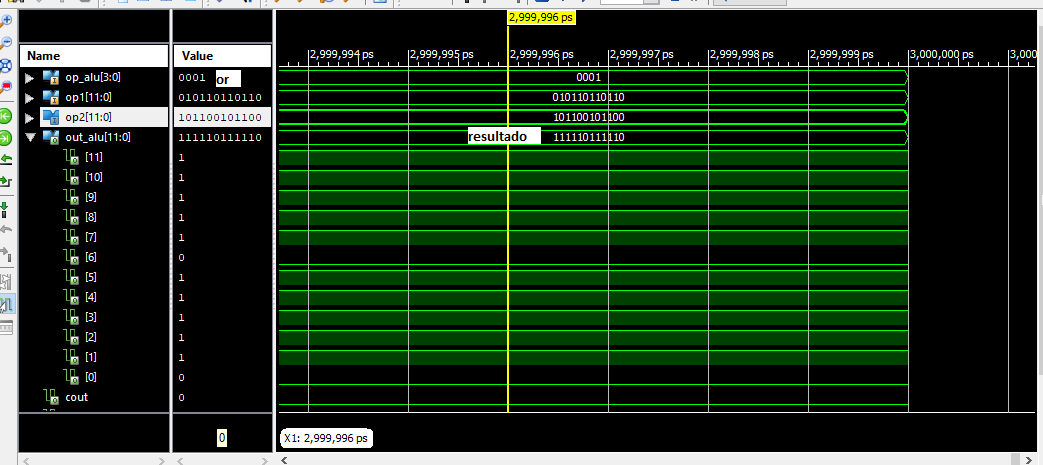


Caso 2 de And

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| OR | 001001110101  OR 000111010110  --------------------  001111110111  VALOR EM HEXA: 3F7 | 010110110110  OR 101100101100  --------------------  111110111110  VALOR EM HEXA: FBE |

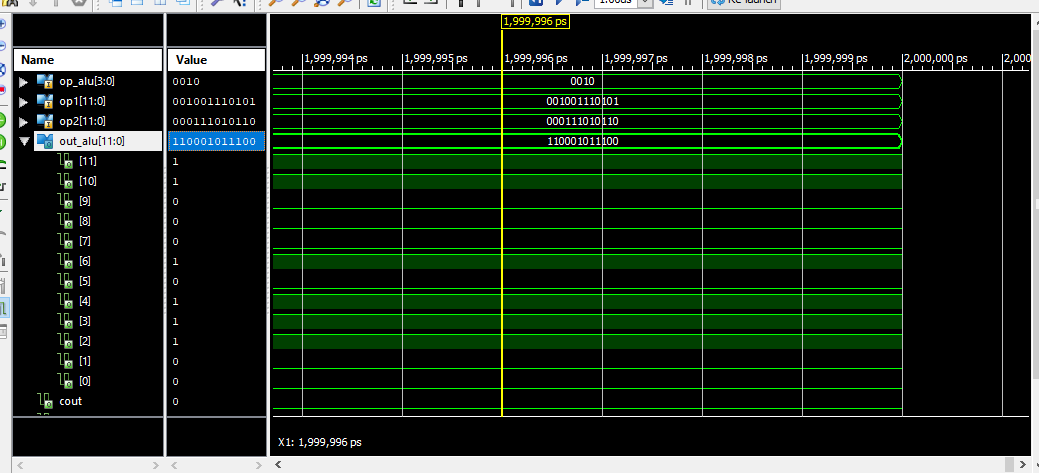


Caso 1 de Or

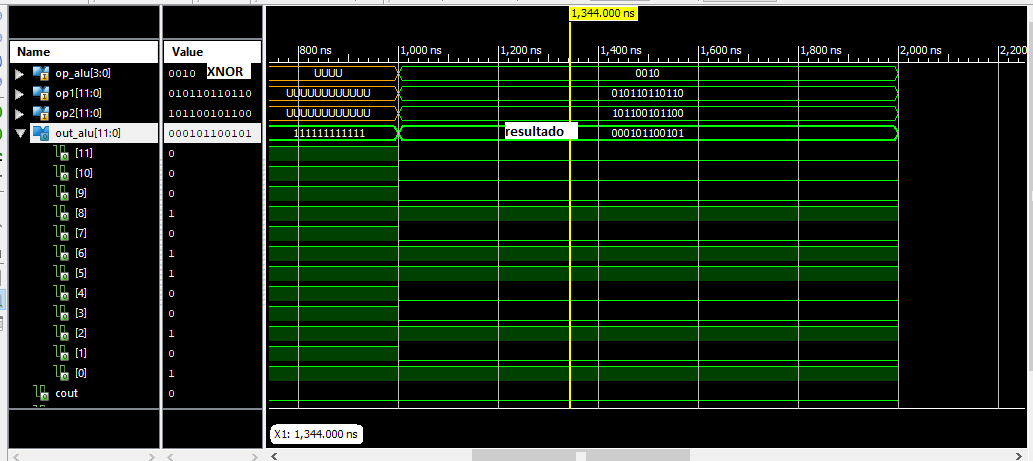


Caso 2 de Or

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

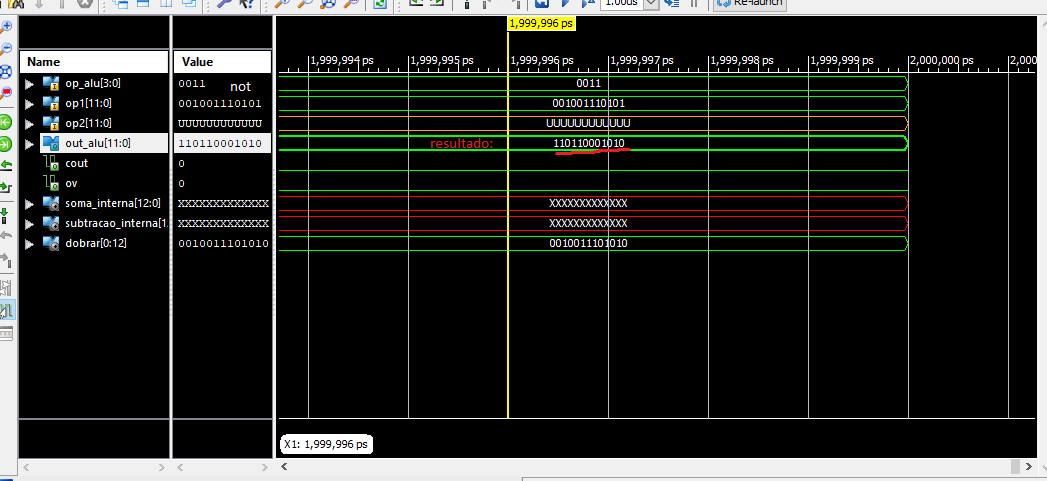


Caso 1 de Xnor

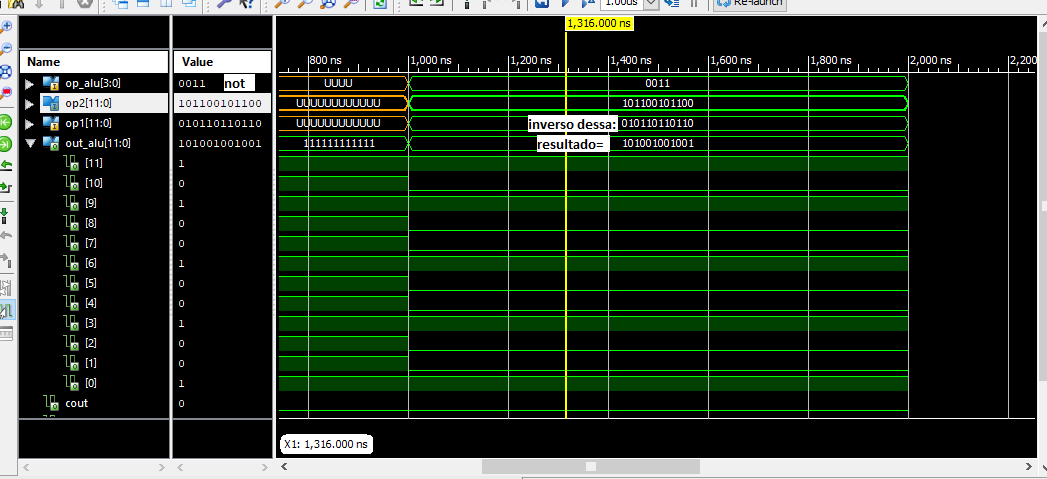


Caso 2 de Xnor

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| NOT | 001001110101  --------------------  110110001010  VALOR EM HEXA: D8A | 010110110110  --------------------  101001001001  VALOR EM HEXA: A49 |

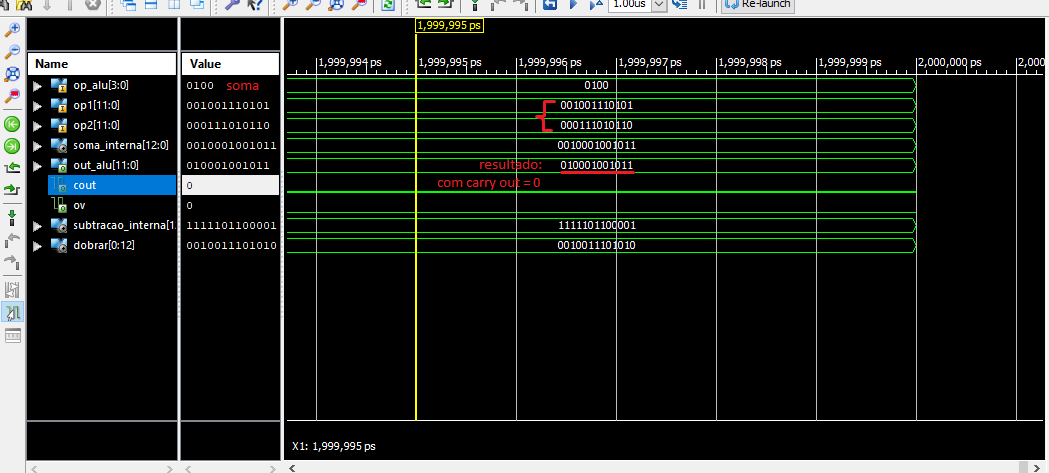


Caso 1 de Not

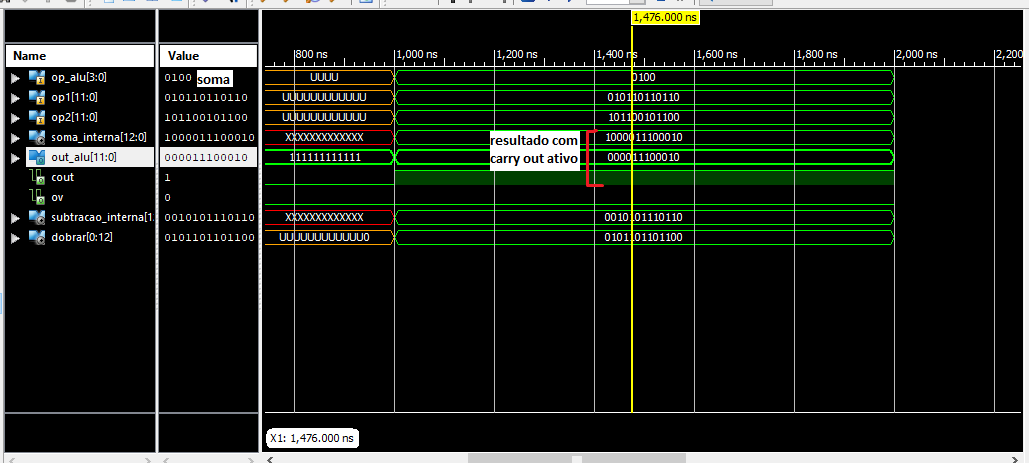


Caso 2 de Not

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| SOMA | 001001110101  + 000111010110  --------------------  010001001011  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 | 010110110110  + 101100101100  --------------------  1 000011100010  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 |
|  |  |  |

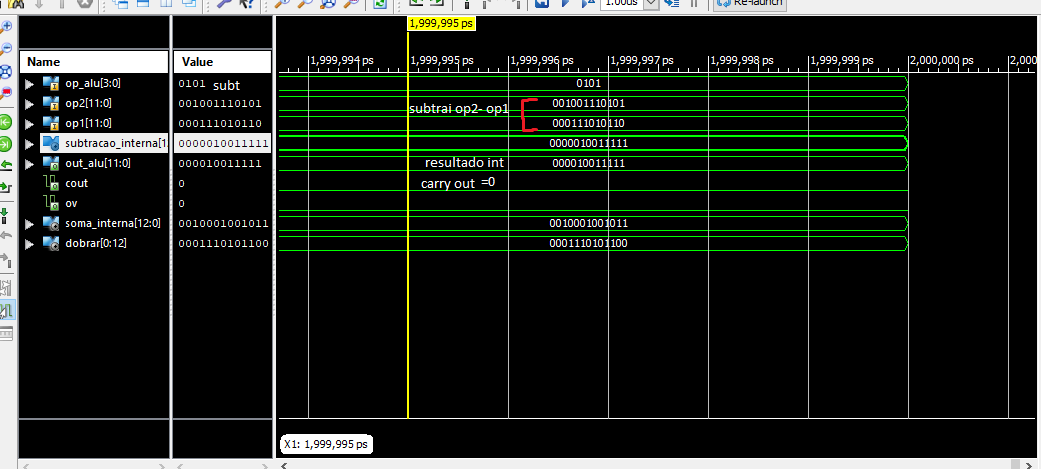


Caso 1 de Soma

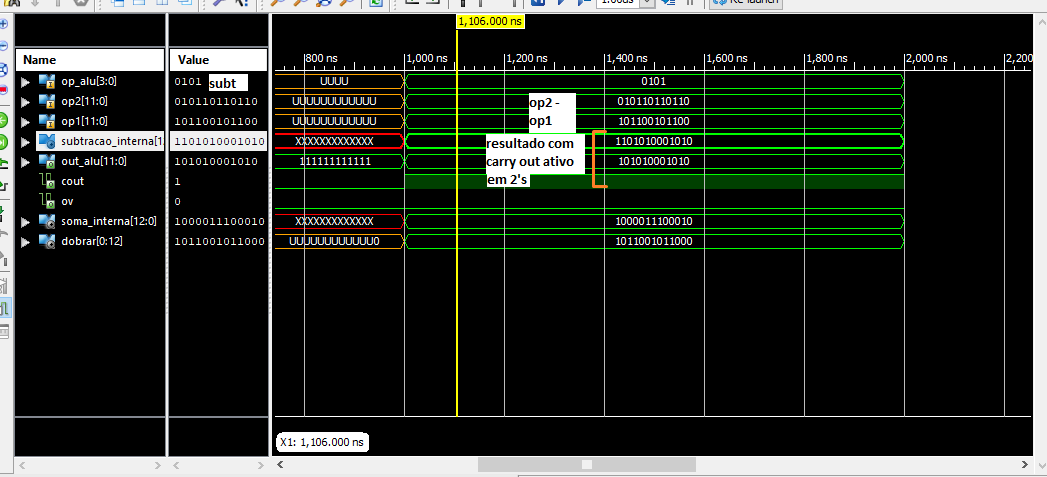


Caso 2 de Soma

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| SUBTRAÇÃO | 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 | 010110110110  - 101100101100  --------------------  1 101010001010  (resultado em complem de dois)  CARRY OUT = 1  VALOR EM HEXA: -576  \*O carry out recebe 1 pois o ultimo bit precisaria “pegar emprestado” 1 bit da próximo fileira, porém essa fileira não é representada devido ao tamanho do vetor |

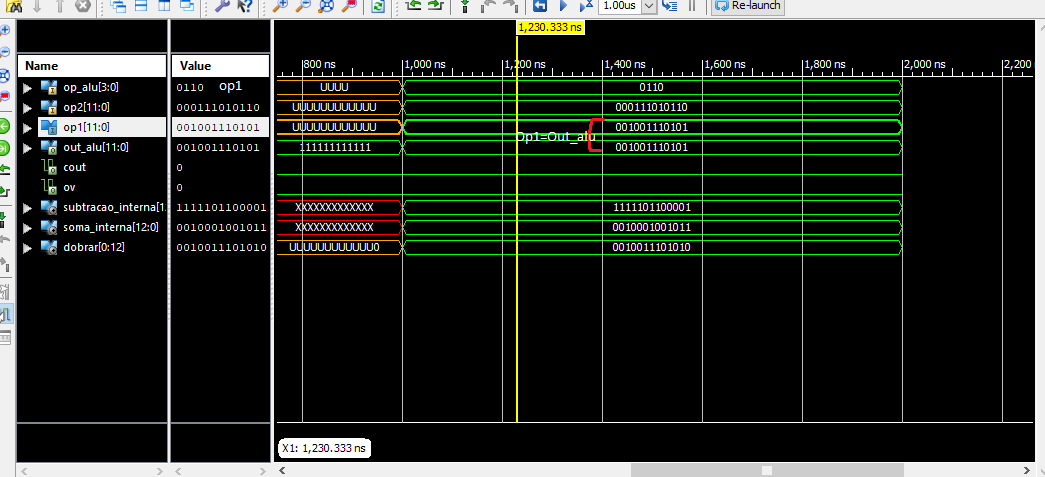


Caso 1 de Subtração

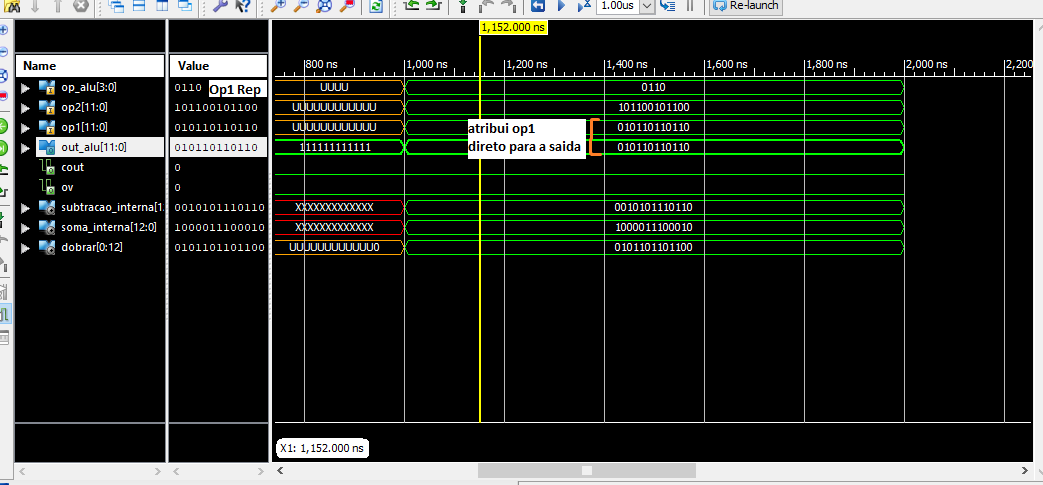


Caso 2 de Subtração

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| OP1 | 001001110101  --------------------  001001110101  VALOR EM HEXA: 275 | 010110110110  --------------------  010110110110  VALOR EM HEXA: 5B6 |

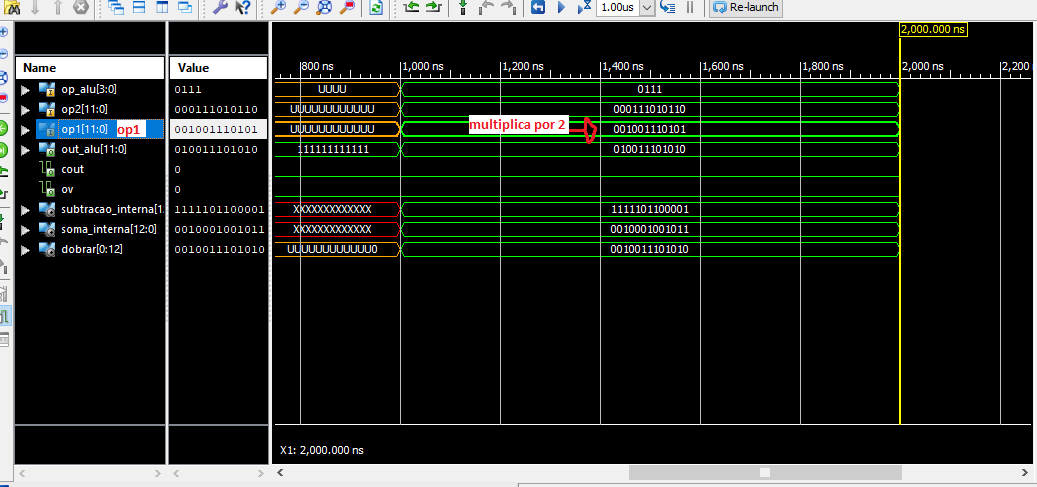


Caso 1 de Op1

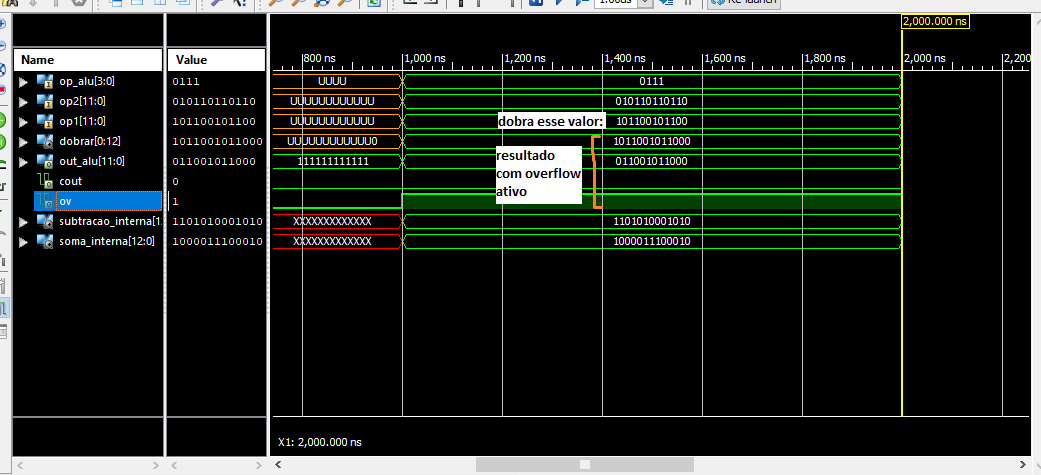


Caso 2 de Op1

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| MU2 | 001001110101  + 001001110101  --------------------  010011101010  OVERFLOW= 0, não ocorre  VALOR EM HEXA: 4EA | 101100101100  + 101100101100  --------------------  1 011001011000  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 |

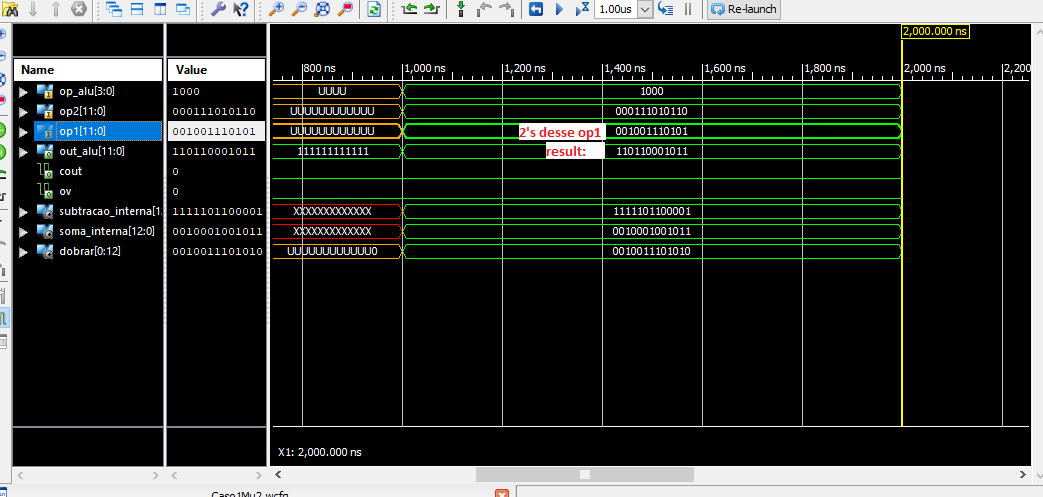


Caso 1 de Mu2

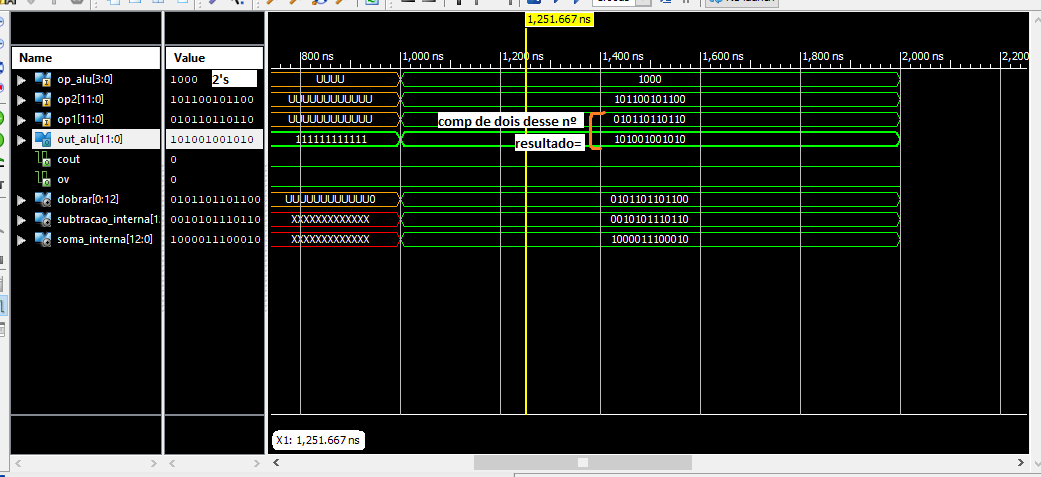


Caso 2 de Mu2

|  |  |  |
| --- | --- | --- |
| operação | Caso 1 | Caso 2 |
| 2’s | 001001110101 (inverte abaixo)  110110001010  + 1  --------------------  110110001011  VALOR EM HEXA: -D8B | 010110110110 (inverte abaixo)  101001001001  + 1  --------------------  101001001010  VALOR EM HEXA: -5B7 |



Caso 1 de Complemento de dois



Caso 2 do Complemento de dois