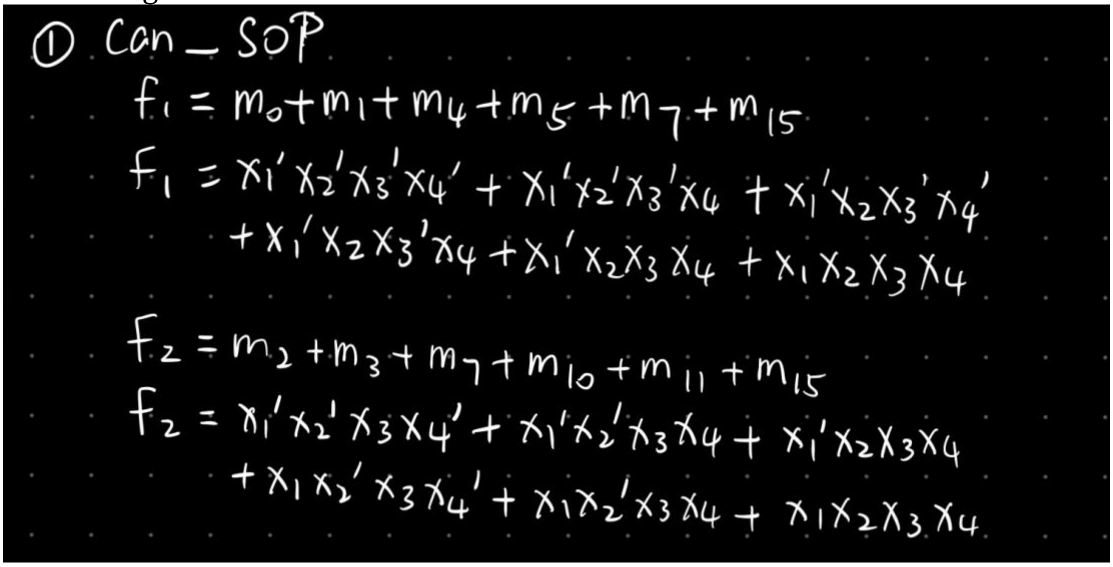
# VHDL Assignment 2

| Student 1 name: | ID number | Student 2 name: | ID number: |
|---|---|---|---|
| Yutong Wang | 260839723 | Hongtao Xu | 260773785 |

1. Explain your VHDL code

- Task 1: As shown below the Canonical SOP expression was derived from the truth table. Then the function was implemented using direct assignment.



① Can_SOP

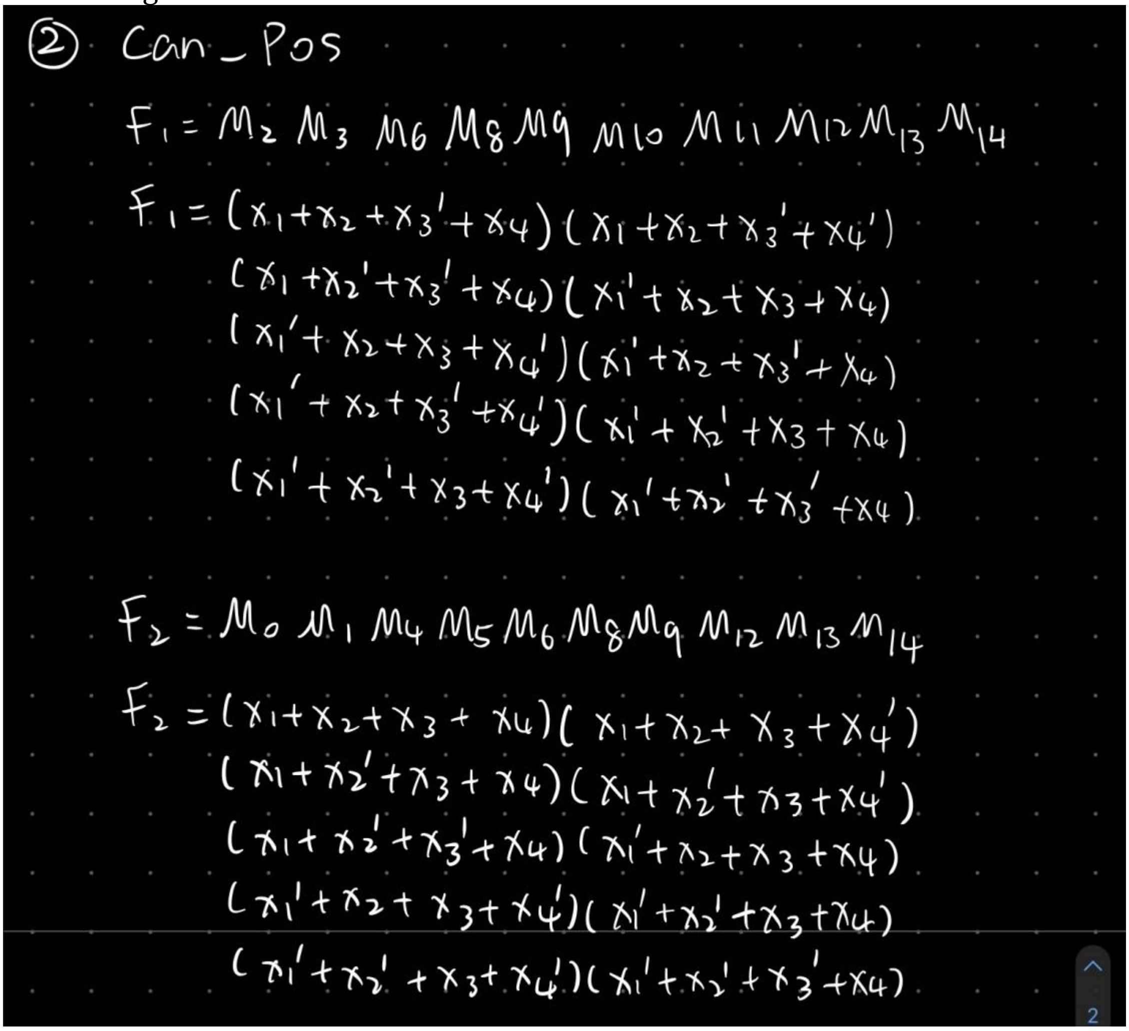
$f_1 = m_0+m_1+m_4+m_5+m_7+m_{15}$

$f_1 = x_1'x_2'x_3'x_4' + x_1'x_2'x_3'x_4 + x_1'x_2x_3'x_4'$
$+ x_1'x_2x_3'x_4 + x_1'x_2x_3x_4 + x_1x_2x_3x_4$

$f_2 = m_2+m_3+m_7+m_{10}+m_{11}+m_{15}$

$f_2 = x_1'x_2'x_3x_4' + x_1'x_2'x_3x_4 + x_1'x_2x_3x_4$
$+ x_1x_2'x_3x_4' + x_1x_2'x_3x_4 + x_1x_2x_3x_4$

- Task 2: As shown below the Canonical POS expression was derived from the truth table. Then the function was implemented using direct assignment.



② Can_Pos

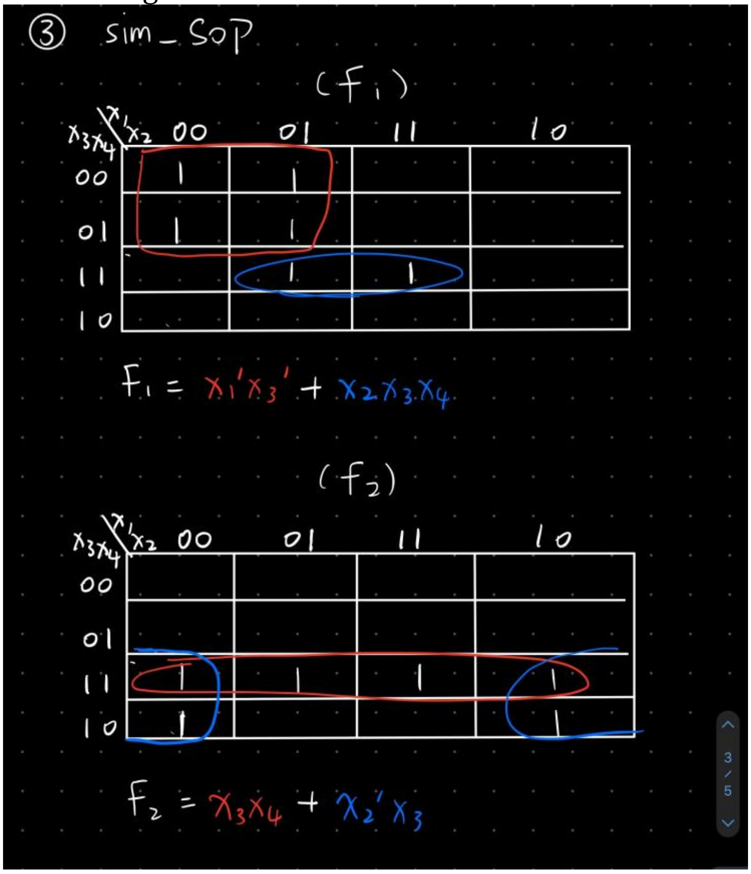
$F_1 = M_2 M_3 M_6 M_8 M_9 M_{10} M_{11} M_{12} M_{13} M_{14}$

$F_1 = (x_1+x_2+x_3'+x_4)(x_1+x_2+x_3'+x_4')$
$(x_1+x_2'+x_3'+x_4)(x_1'+x_2+x_3+x_4)$
$(x_1'+x_2+x_3+x_4')(x_1'+x_2+x_3'+x_4)$
$(x_1'+x_2+x_3'+x_4')(x_1'+x_2'+x_3+x_4)$
$(x_1'+x_2'+x_3+x_4')(x_1'+x_2'+x_3'+x_4)$

$F_2 = M_0 M_1 M_4 M_5 M_6 M_8 M_9 M_{12} M_{13} M_{14}$

$F_2 = (x_1+x_2+x_3+x_4)(x_1+x_2+x_3+x_4')$
$(x_1+x_2'+x_3+x_4)(x_1+x_2'+x_3+x_4')$
$(x_1+x_2'+x_3'+x_4)(x_1'+x_2+x_3+x_4)$
$(x_1'+x_2+x_3+x_4')(x_1'+x_2'+x_3+x_4)$
$(x_1'+x_2'+x_3+x_4')(x_1'+x_2'+x_3'+x_4)$

- Task 3: As shown below, K-map was used to obtain the simplified expression of SOP. Then the function was implemented using direct assignment.



③ sim_SoP

($f_1$)

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | | |
| 01 | 1 | 1 | | |
| 11 | | 1 | 1 | |
| 10 | | | | |

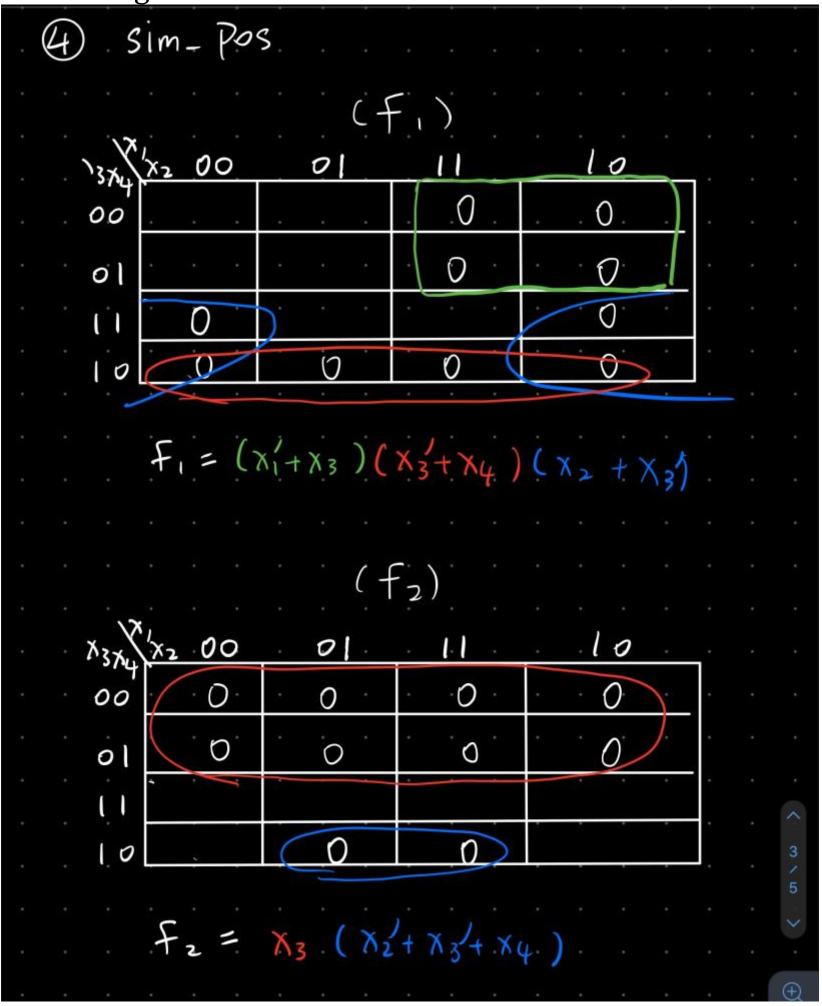
$F_1 = x_1'x_3' + x_2x_3x_4$

($f_2$)

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | | | 1 |

$F_2 = x_3x_4 + x_2'x_3$

- Task 4: As shown below, K-map was used to obtain the simplified expression of POS. Then the function was implemented using direct assignment



④ sim_Pos

($f_1$)

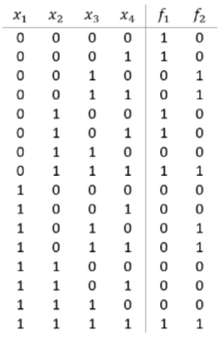
| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 0 | 0 |
| 01 | | | 0 | 0 |
| 11 | 0 | | | 0 |
| 10 | 0 | 0 | 0 | 0 |

$F_1 = (x_1'+x_3)(x_3'+x_4)(x_2+x_3')$

($f_2$)

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | | | | |
| 10 | | 0 | 0 | |

$F_2 = x_3\ (x_2'+x_3'+x_4)$

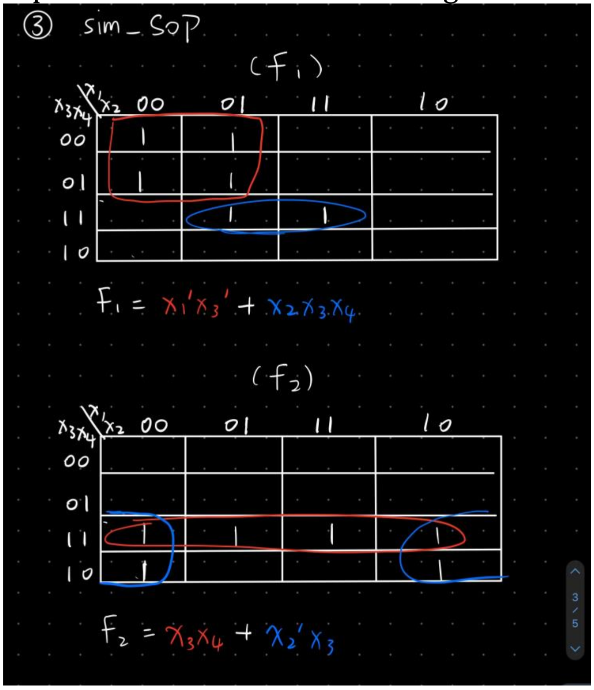


| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f_1$ | $f_2$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

- Task 5: From the truth table, there are 4 times of input values generating output f1f2 as "10": row 0,1,4 and 5. And 4 times of input generates output as "01": row 2,3,10 and 11. Also, 2 times generating "11": row 6 and 15. The remaining 6 rows generate output  as "00".
  By using Select/With statement, we can group those inputs which generates the same combination of output respectively and quickly. It is the value of control expression compared with n possible options, "when" a match is found, the value for the option would be assigned.

- Task 6: From Task 5, we know that there 4 groups of inputs generating the same combination of output. This time we use When/Else statement to group them. It is different from groups combination, which means that each arrangement of input has its own value to correspond. This is the expressions after "when" are evaluated consistently until a true expression found.  The last assignment would be evaluated if none of true expressions found before. The earlier evaluated expression has a higher priority.

- Task 7: We found that the most joint-optimized SOP to represent F1 and F2 is the arrangement we have on Task3:



③ sim_SOP

(f1)

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | | |
| 01 | 1 | 1 | | |
| 11 | | 1 | 1 | |
| 10 | | | | |

$F_1 = x_1'x_3' + x_2x_3x_4$

(f2)

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | | | 1 |

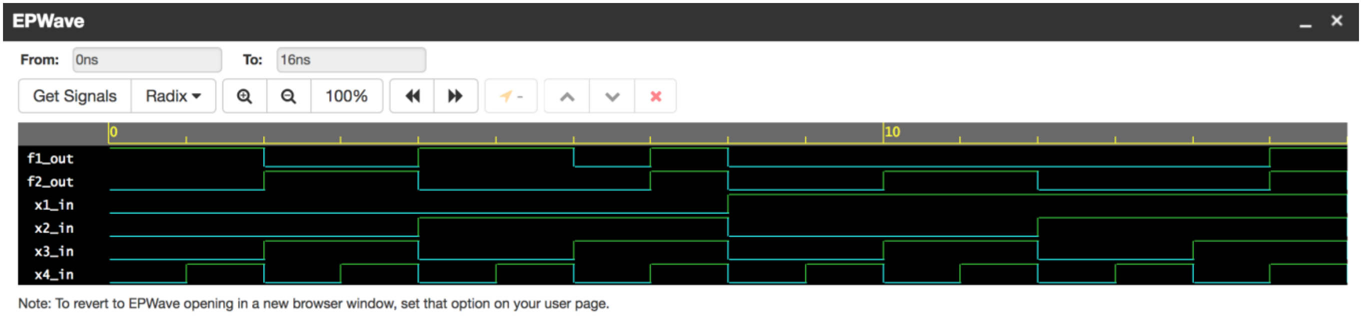
$F_2 = x_3x_4 + x_2'x_3$

2.

| Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Cost | 82 | 82 | 24 | 24 | N/A | N/A | 24 |
| Logic Utilization (in LUTs) | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 | Used:2<br>Avail:63400 |

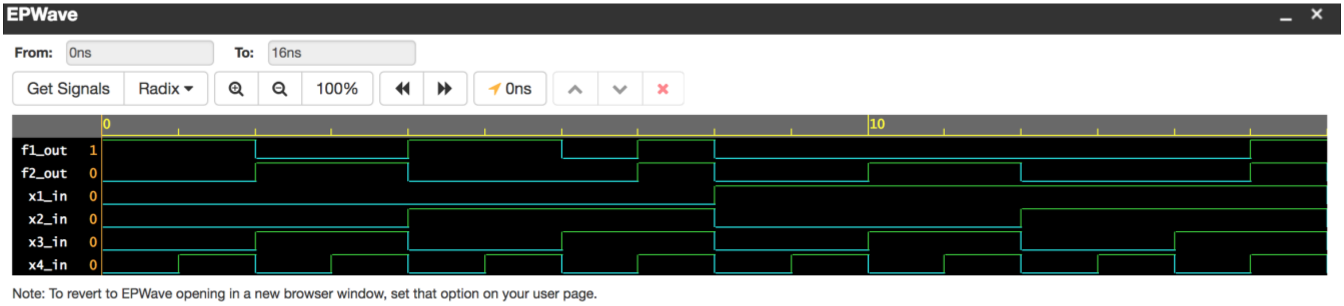
3. Show representative simulation plots of all tasks for all the possible input values (exhaustive test results).

- Task 1 can_SOP:



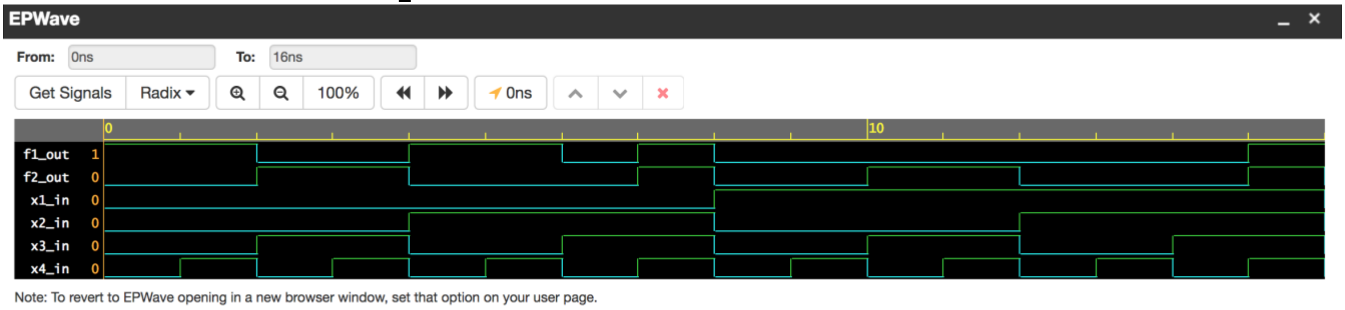
Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- Task 2 can_POS:



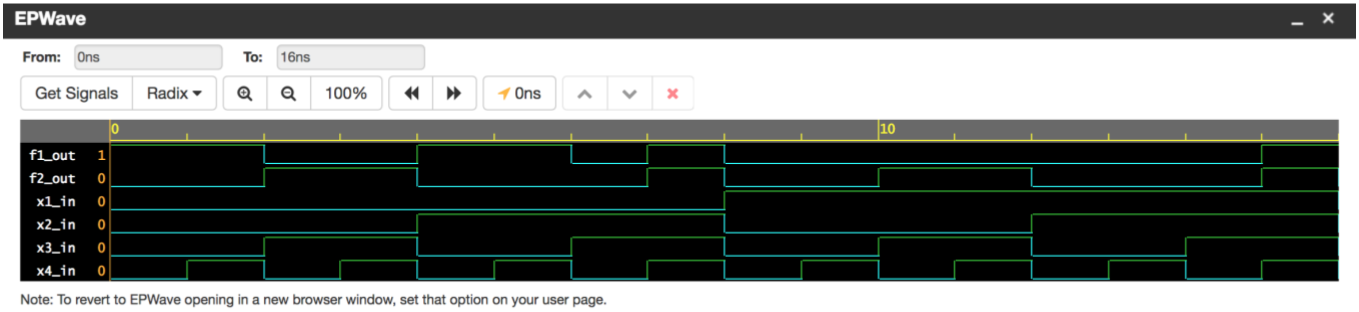
Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- Task 3 sim_SOP:



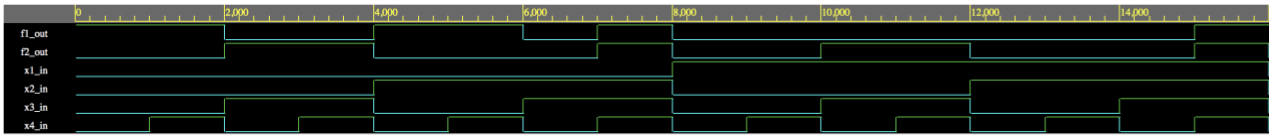
Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- Task 4 sim_POS:

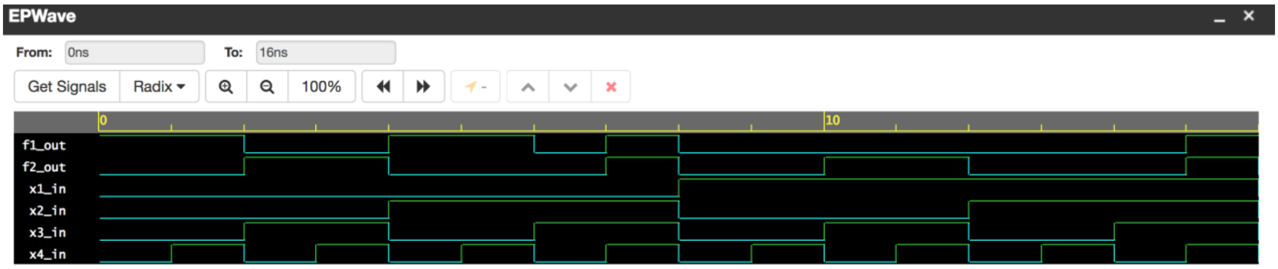


Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- Task 5 select_assignment:



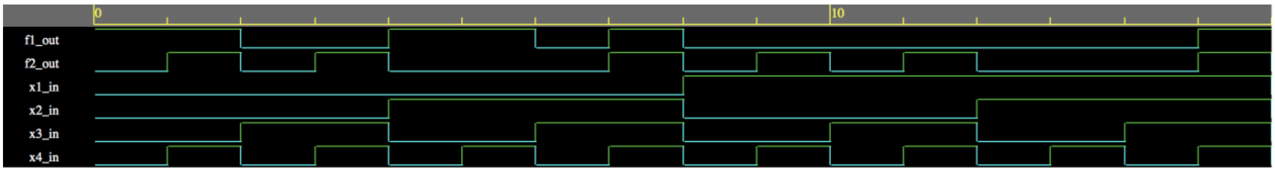
- Task 6 when_assignment:



From: 0ns To: 16ns

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- Task 7 joint_SOP:


0
10
f1_out
f2_out
x1_in
x2_in
x3_in
x4_in