

27. Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2_REG, que implemente a função $B = (A * 2) - (A \% 4)$. Apresentar a computação e a função computada para as entradas 4 e 6.

```
R1: Se a_zero então vá_para Rx senão vá_para R2;
R2: Faça subtrair_a vá_para R3;
R3: Faça adicionar_b vá_para R4;
R4: Se a_zero então vá_para Rx senão vá_para R5;
R5: Faça subtrair_a vá_para R6;
R6: Faça adicionar_b vá_para R7;
R7: Se a_zero então vá_para Rx senão vá_para R8;
R8: Faça subtrair_a vá_para R9;
R9: Faça adicionar_b vá_para R10;
R10: Se a_zero então vá_para Rx senão vá_para R11;
R11: Faça subtrair_a vá_para R12;
R12: Faça adicionar_b vá_para R13;
R13: Faça adicionar_b vá_para R14;
R14: Faça adicionar_b vá_para R15;
R15: Faça adicionar_b vá_para R16;
R16: Faça adicionar_b vá_para R1;
```

```
(R1 , (4, 0))
(R2 , (4, 0))
(R3 , (3, 0))
(R4 , (3, 1))
(R5 , (3, 1))
(R6 , (2, 1))
(R7 , (2, 2))
(R8 , (2, 2))
(R9 , (1, 2))
(R10, (1, 3))
(R11, (1, 3))
(R12, (0, 3))
(R13, (0, 4))
(R14, (0, 5))
(R15, (0, 6))
(R16, (0, 7))
(R1 , (0, 8))
(RX , (0, 8))
```

<PROG, 2_REG> : 4 -> 8

```
(R1 , (6, 0))
(R2 , (6, 0))
(R3 , (5, 0))
(R4 , (5, 1))
(R5 , (5, 1))
(R6 , (4, 1))
(R7 , (4, 2))
(R8 , (4, 2))
(R9 , (3, 2))
(R10, (3, 3))
(R11, (3, 3))
(R12, (2, 3))
(R13, (2, 4))
(R14, (2, 5))
(R15, (2, 6))
(R16, (2, 7))
(R1 , (2, 8))
(R2 , (2, 8))
(R3 , (1, 8))
(R4 , (1, 9))
(R5 , (1, 9))
(R6 , (0, 9))
(R7 , (0, 10))
(RX , (0, 10))
```

<PROG, 2_REG> : 6 -> 10