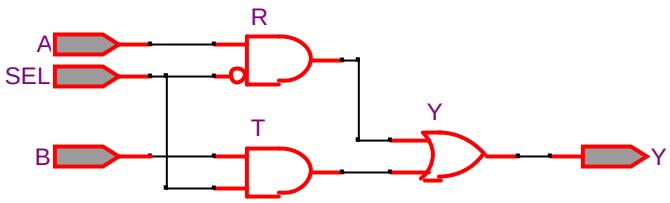
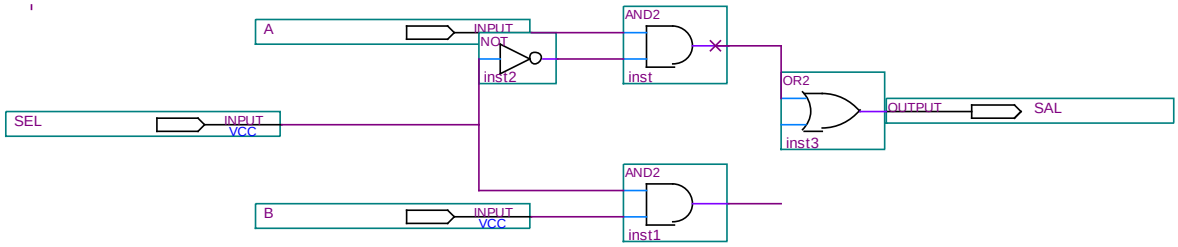
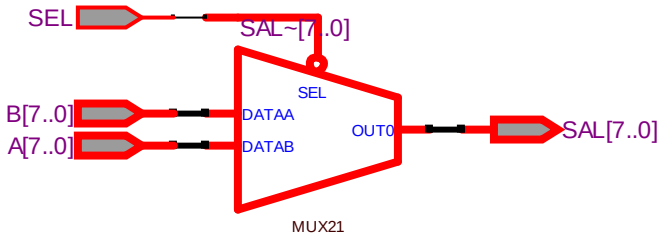
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY MUX2X1 IS
PORT(A,B,SEL: IN STD_LOGIC;
        Y: OUT STD_LOGIC);
END ENTITY;
ARCHITECTURE MUX2X1 OF MUX2X1 IS
        SIGNAL R,T: STD_LOGIC;
        BEGIN
                R<=A AND (NOT SEL);
                T<=B AND SEL;
                Y<=R OR T;
END ARCHITECTURE;
```




A
INPUT
NOT
inst2
AND2
inst
OR2
inst3
SEL
INPUT
VCC
OUTPUT
SAL
AND2
B
INPUT
VCC
inst1

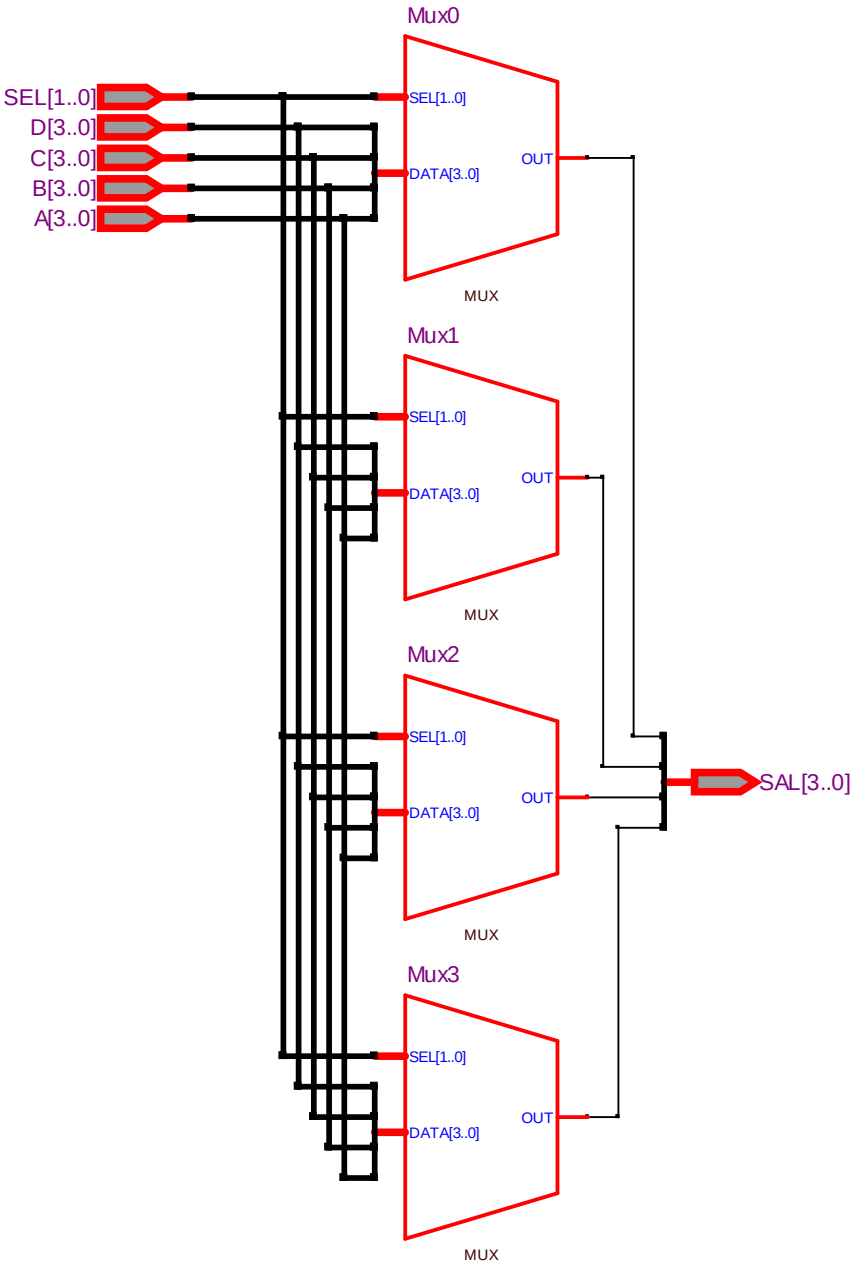
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY MUX2X1_8BITS IS
	PORT (A,B: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
		SEL: IN STD_LOGIC;
		SAL: OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END ENTITY;
ARCHITECTURE MEX2X1_8BITS OF MUX2X1_8BITS IS
	BEGIN
	SAL<=A WHEN SEL='0' ELSE B;
END ARCHITECTURE;
```



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
--TIPO VECTOR
--ENTITY MUX4X1_4BITS IS
--	PORT(A,B,C,D: IN STD_LOGIC_VECTOR (3 DOWNTO 0);
--	SAL: OUT STD_LOGIC_VECTOR (3 DOWNTO 0);
--	SEL: IN STD_LOGIC_VECTOR (1 DOWNTO 0));
--END ENTITY;
--ARCHITECTURE MUX4X1_4BITD OF MUX4X1_4BITS IS
--	BEGIN
--		SAL<=A WHEN SEL="00" ELSE
--		B WHEN SEL="01" ELSE
--		C WHEN SEL="10" ELSE
--		D;
--END ARCHITECTURE;

--SEÑALES TIPO ENTERO
ENTITY MUX4X1_4BITS IS
	PORT(A,B,C,D: IN INTEGER RANGE 0 TO 15;
	SAL: OUT INTEGER RANGE 0 TO 15;
	SEL: IN INTEGER RANGE 0 TO 3);
END ENTITY;
ARCHITECTURE MUX4X1_4BITS OF MUX4X1_4BITS IS
	BEGIN
		WITH SEL SELECT
			SAL<=A WHEN 0,
			B WHEN 1,
			C WHEN 2,
			D WHEN OTHERS;
```

END ARCHITECTURE;



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY MUX4X1 IS
PORT(E3,E2,E1,E0,SEL1,SEL0: IN STD_LOGIC;
        Y: OUT STD_LOGIC);
END ENTITY;
ARCHITECTURE MUX4X1 OF MUX4X1 IS
        SIGNAL R,T,U,V: STD_LOGIC;
        BEGIN
                R<=E3 AND SEL1 AND SEL0;
                T<=E2 AND SEL1 AND (NOT SEL0);
                U<=E1 AND (NOT SEL1) AND SEL0;
                V<=E0 AND (NOT SEL1) AND (NOT SEL0);
                Y<=R OR T OR U OR V;
END ARCHITECTURE;
```