

```

LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

USE ieee.std_logic_signed.ALL;


ENTITY lab5 IS

    GENERIC (x : INTEGER := 6;
             y : INTEGER := 6);

    PORT(m : IN STD_LOGIC_VECTOR(x - 1 DOWNT0 0);
         r : IN STD_LOGIC_VECTOR(y - 1 DOWNT0 0);
         result : OUT STD_LOGIC_VECTOR(x + y - 1 DOWNT0 0));

END lab5;


ARCHITECTURE behav OF lab5 IS

    BEGIN

        PROCESS(m, r)

            CONSTANT X_ZEROS : STD_LOGIC_VECTOR(x - 1 DOWNT0 0) := (OTHERS => '0');
            CONSTANT Y_ZEROS : STD_LOGIC_VECTOR(y - 1 DOWNT0 0) := (OTHERS => '0');

            VARIABLE a, s, p : STD_LOGIC_VECTOR(x + y + 1 DOWNT0 0);
            VARIABLE mn : STD_LOGIC_VECTOR(x - 1 DOWNT0 0);

            BEGIN

```

```
a := (OTHERS => '0');
```

```
s := (OTHERS => '0');
```

```
p := (OTHERS => '0');
```

```
IF (m /= X_ZEROS AND r /= Y_ZEROS) THEN -- not equal to 0
```

```
a(x + y DOWNT0 y + 1) := m; -- adder
```

```
a(x + y + 1) := m(x - 1); -- signed bit
```

```
mn := (NOT m) + 1; -- 2's
```

```
s(x + y DOWNT0 y + 1) := mn; -- adder for negative #
```

```
s(x + y + 1) := NOT(m(x - 1)); -- signed bit
```

```
p(y DOWNT0 1) := r; -- set product
```

```
FOR i IN 1 TO y LOOP --booth encode
```

```
  IF (p(1 DOWNT0 0) = "01") THEN
```

```
    p := p + a;
```

```
  ELSIF (p(1 DOWNT0 0) = "10") THEN
```

```
    p := p + s;
```

```
  END IF;
```

```
  p(x + y DOWNT0 0) := p(x + y + 1 DOWNT0 1); -- right shift
```

```
END LOOP;
```

END IF;

result <= p(x + y DOWNT0 1); -- set result

END PROCESS;

END behav;

