COL215 Digital Logic and System Design

VHDL – concurrent and sequential assignments

16 August 2017

Outline

- Concurrent statements
- Sequential statements
- Types – bit vectors
- 3-port switch example

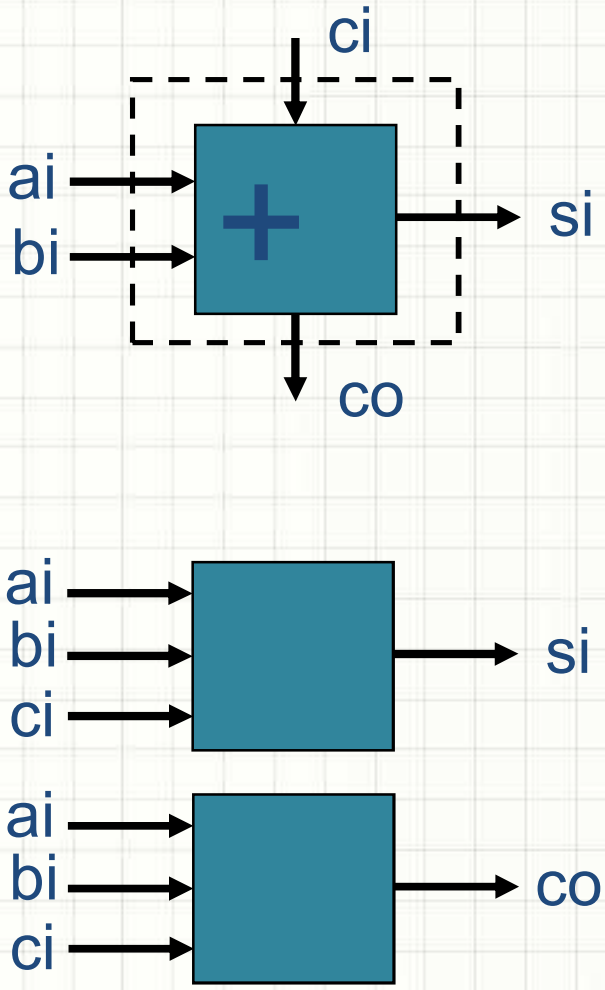
Concurrent assignments

```
ARCHITECTURE data_flow
OF full_adder IS

BEGIN
 si <= ai XOR bi XOR ci;

 co <= (ai AND bi) OR (bi AND ci)
          OR (ai AND ci);
END data_flow;
```

Concurrent Signal Assignments



Is the order significant?

```
ARCHITECTURE data_flow1 OF full_adder IS
BEGIN
  si <= ai XOR bi XOR ci;
  co <= (ai AND bi) OR (bi AND ci) OR (ai AND ci);
END data_flow1;
```

```
ARCHITECTURE data_flow2 OF full_adder IS
BEGIN
  co <= (ai AND bi) OR (bi AND ci) OR (ai AND ci);
  si <= ai XOR bi XOR ci;
END data_flow2;
```

Ordering of dependent assignments

```
ARCHITECTURE arch1 OF e IS
BEGIN
  t <= a OR b;
  s <= (t AND c) OR (NOT a AND d);
END arch1;
```

```
ARCHITECTURE arch2 OF e IS
BEGIN
  s <= (t AND c) OR (NOT a AND d);
  t <= a OR b;
END arch2;
```

Example with internal signal

```
ENTITY comb_logic IS
PORT (i1, i2, i3, i4: IN BIT;
        o1, o2: OUT BIT);
END comb_logic;

ARCHITECTURE data_flow1
OF comb_logic IS

BEGIN

  o1 <= (i1 and i2 and i3) xor i2;
  o2 <= (i1 and i2 and i3) or i4;
END data_flow1;
```

in any order

```
ENTITY comb_logic IS
PORT (i1, i2, i3, i4: IN BIT;
        o1, o2: OUT BIT);
END comb_logic;

ARCHITECTURE data_flow2
OF comb_logic IS
SIGNAL temp: BIT;
BEGIN
  temp <= (i1 and i2 and i3);
  o1 <= temp xor i2;
  o2 <= temp or i4;
END data_flow2;
```

avoid repeated evaluation of common sub-expression

Procedural description

```
ENTITY maj3 IS
  PORT (a, b, c : IN BIT;
                y       : OUT BIT);
END maj3;
ARCHITECTURE sequential OF maj3 IS
BEGIN
  PROCESS (a, b, c)
  BEGIN
    y <= (a AND b) OR (b AND c) OR (a AND c);
  END PROCESS;
END ARCHITECTURE sequential;
```

Sensitivity list

What happens if all inputs are not included in the sensitivity list?

Combinational Circuit

```
ENTITY maj3 IS
   PORT (a, b, c : IN BIT;
                y        : OUT BIT);
END maj3;
```

```
ARCHITECTURE conc
   OF maj3 IS
BEGIN
   y <= (a AND b) OR
         (b AND c) OR
         (a AND c);
END conc;
```

```
ARCHITECTURE seq
   OF maj3 IS
BEGIN
   PROCESS (a, b, c)
   BEGIN
      y <= (a AND b) OR
            (b AND c) OR
            (a AND c);
   END PROCESS;
END seq:
```

Architecture: General form

```
ARCHITECTURE name OF entity_name IS
  declarations
  BEGIN
    concurrent statement
    concurrent statement
    . . .
    concurrent statement
  END ARCHITECTURE name;
```

Process: General form

```
PROCESS (sensitivity list)
  declarations
  BEGIN
    sequential statement
    sequential statement
    . . .
    sequential statement
  END PROCESS;
```

Types in VHDL

- Scalar
  - enumeration
  - integer
  - floating point
  - physical
- Subtypes
  - index constraint
  - range constraint
- Composite
  - arrays
  - records
- Others (access type, file type)

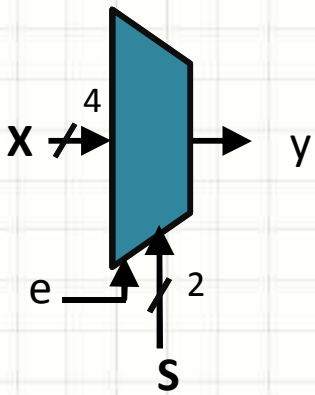
Types in Standard Package

- type boolean is (false, true);
- type bit is (‘0’ , ‘1’);
- type character is (…, ‘0’ , ‘1’ , …, ‘A’ , ‘B’ , …, ‘a’ , ‘b’ , …);
- type integer is range -2147483647 to 2147483647;
- type real is …;
- type time is …;
- type natural is integer range 0 to integer’ high;
- type bit_vector is array (natural range <>) of bit;
- ……

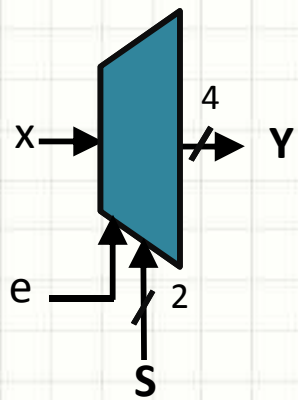
bit_vector example

```
ENTITY aCircuit IS
 PORT (av, cv : IN bit_vector (7 DOWNTO 0);
          w : OUT bit;
          wv : OUT bit_vector (7 DOWNTO 0)
         );
END ENTITY aCircuit;
ARCHITECTURE indexing_slicing OF aCircuit IS
    SIGNAL dv : bit_vector (7 DOWNTO 0);
BEGIN
    dv <= av AND cv;
    wv (3 DOWNTO 0) <= dv (7 DOWNTO 4) OR
                                dv (3 DOWNTO 0);
    w <= cv (4);
END ARCHITECTURE indexing_slicing ;
```

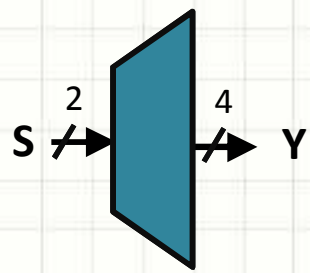
Combinational Modules



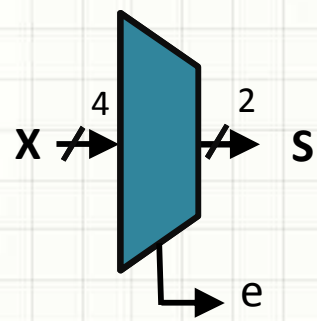
4:1 mux



1:4 de-mux



2:4 decoder



4 input priority encoder

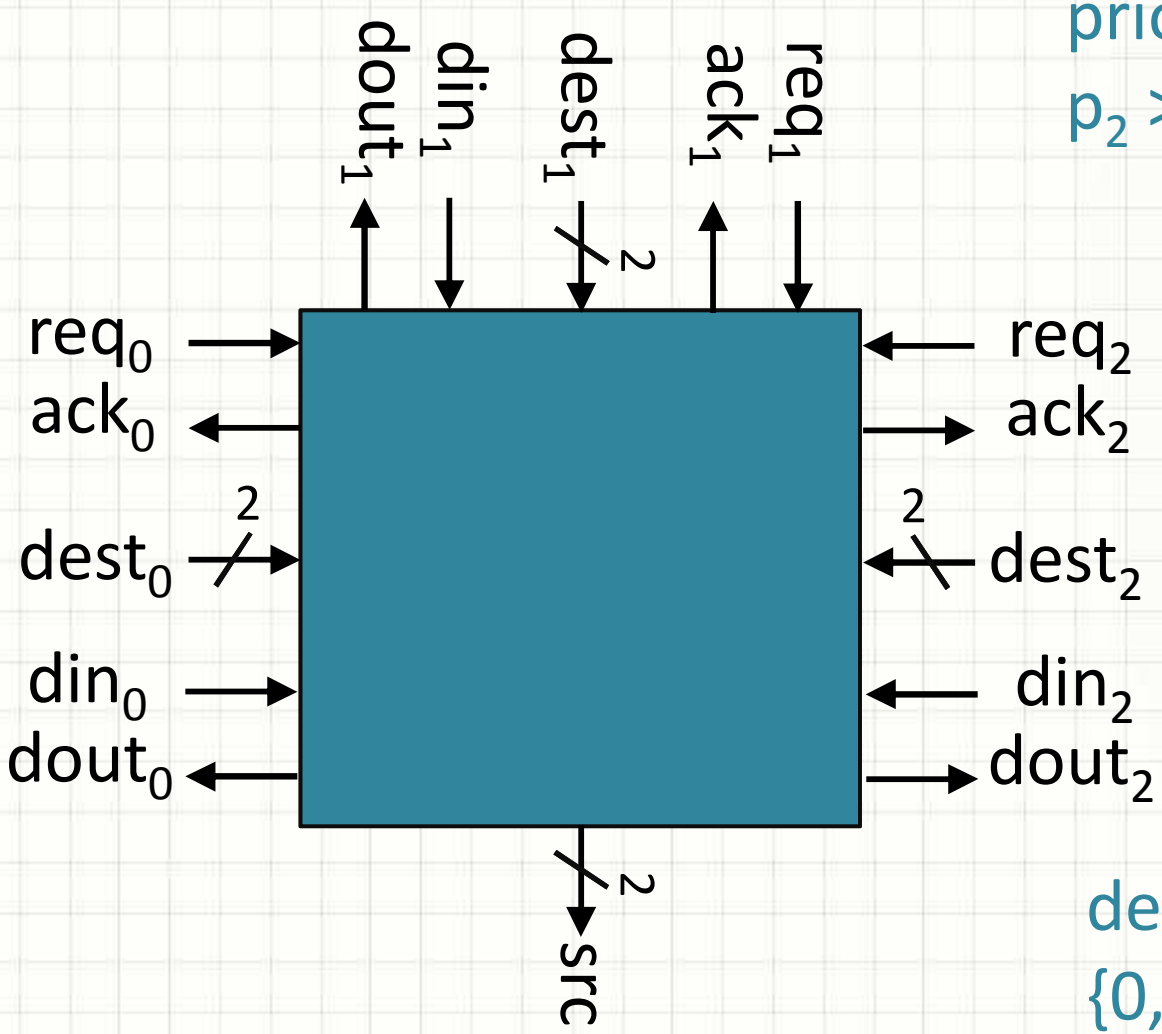
e	s_1s_0	y
0	- -	0
1	0 0	x_0
1	0 1	x_1
1	1 0	x_2
1	1 1	x_3

e	s_1s_0	$y_3y_2y_1y_0$
0	- -	0 0 0 0
1	0 0	0 0 0 x
1	0 1	0 0 x 0
1	1 0	0 x 0 0
1	1 1	x 0 0 0

s_1s_0	$y_3y_2y_1y_0$
0 0	0 0 0 1
0 1	0 0 1 0
1 0	0 1 0 0
1 1	1 0 0 0

$x_3x_2x_1x_0$	e s_1s_0
0 0 0 0	0 - -
0 0 0 1	1 0 0
0 0 1 -	1 0 1
0 1 - -	1 1 0
1 - - -	1 1 1

Lab exercise 2 : 3-Port Switch



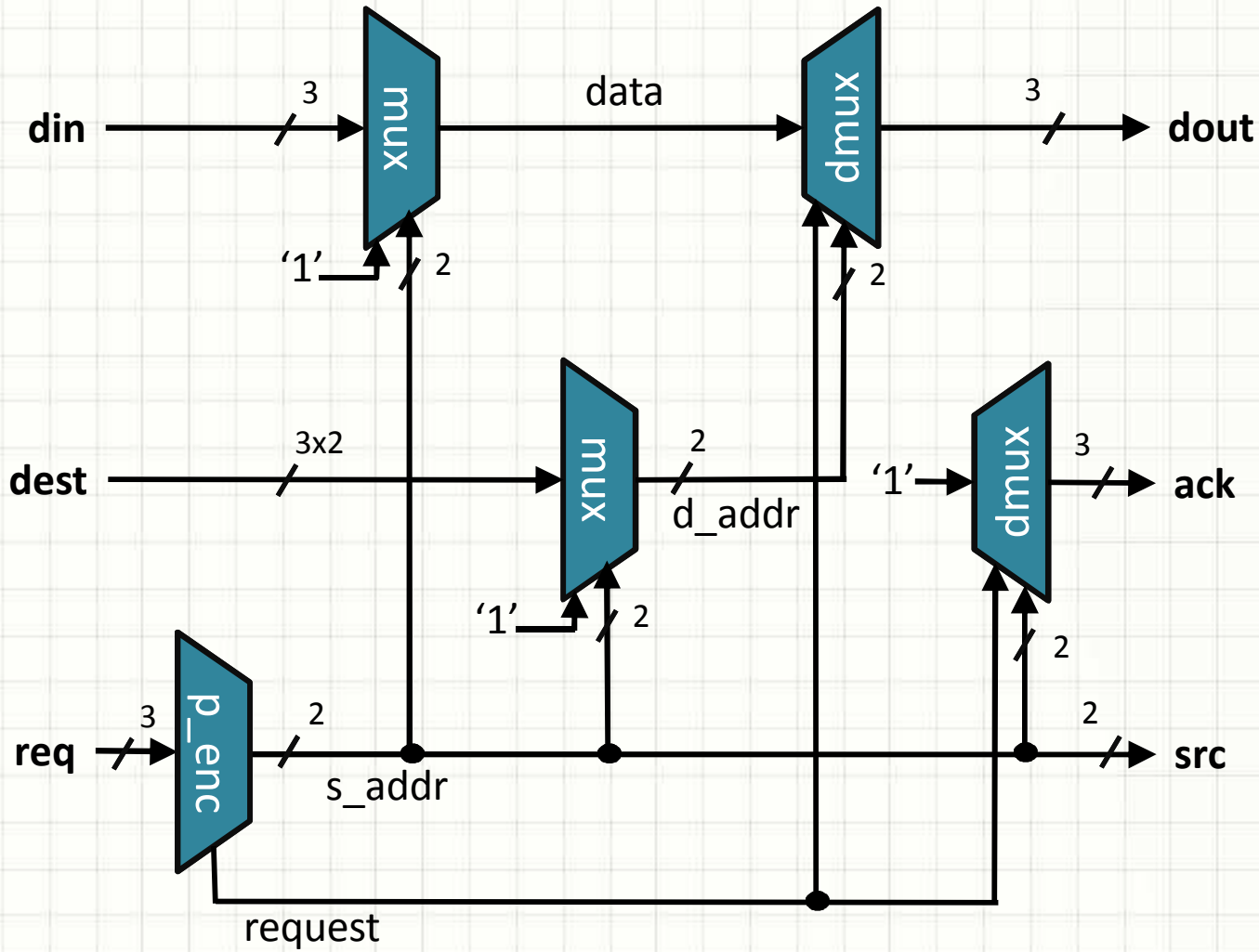
priority:

$p_2 > p_1 > p_0$

$dest_i$, src:

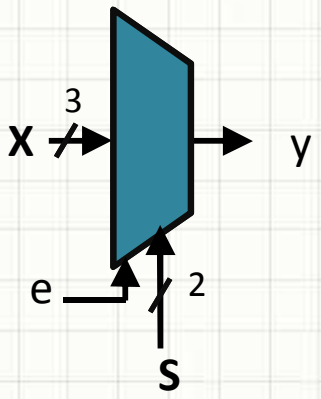
{0, 1, 2}

3-Port Switch Design



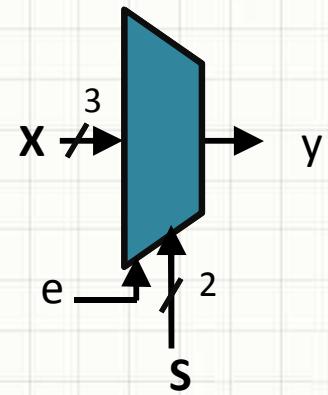
3 : 1 Mux

```
ENTITY mux_3_1 IS
  PORT (X: IN     bit_vector (2 DOWNTO 0);
        S: IN     bit_vector (1 DOWNTO 0);
        e: IN      bit;
        y: OUT bit
       );
END mux_3_1;
```



CASE statement

```
ARCHITECTURE casestmt OF mux_3_1 IS
BEGIN
 PROCESS (S, e)
   BEGIN
     IF e = '1' THEN
       CASE S IS
         WHEN "00"        => y <= X(0);
         WHEN "01"        => y <= X(1);
         WHEN OTHERS  => y <= X(2);
       END CASE;
     ELSE y <= '0';
     ENDIF;
   END PROCESS;
END ARCHITECTURE casestmt;
```

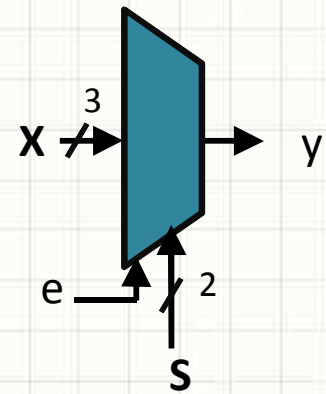


3:1 mux

e	$s_1 s_0$	y
0	- -	0
1	0 0	x_0
1	0 1	x_1
1	1 0	x_2
1	1 1	??

Selected Signal Assignment

```
ARCHITECTURE ssa OF mux_3_1 IS
BEGIN
  SIGNAL t : bit;
  WITH S SELECT
    t <= X(0) WHEN "00",
         X(1) WHEN "01",
         X(2) WHEN OTHERS;
  y <= t AND e;
END ARCHITECTURE ssa;
```



3:1 mux

e	$s_1 s_0$	y
0	- -	0
1	0 0	x_0
1	0 1	x_1
1	1 0	x_2
1	1 1	??

THANKS