Digital Systems - 2

Pere Palà - Alexis López

iTIC http://itic.cat

February 2016

Numeric Basics

- Coding of numeric information:
- Decimal system: 37_{10} means $3 \times 10^1 + 7 \times 10^0$.
- Decimal system: 203_{10} means $2 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$.
- $37_{10} = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$.
- $37_{10} = 100101_2$.

- n bits : $0 \cdots 2^n - 1$.
- To represent $0 \cdots N - 1$ we need $\log_2 N$ bits.

Unsigned Integers

- Declaration: `signal s_unsig : unsigned(3 downto 0);`

Assignments

Legal: `s_unsig <= "1110";`

Illegal: `s_unsig <= 0;`

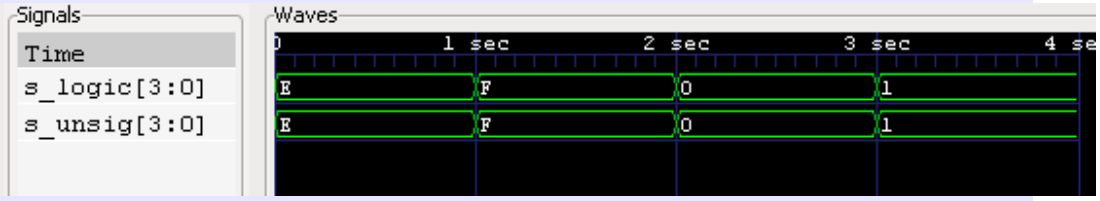
Legal: `s_unsig_dest <= s_unsig+1;`

Conversions

- `my_slv <= std_logic_vector(my_unsigned);`
- `my_unsigned <= unsigned(my_slv);`

A VHDL type test example

```
library ieee; use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity num_tb is
end num_tb ;

architecture behav of num_tb is
  signal s_logic : std_logic_vector(3 downto 0) ;
  signal s_unsig : unsigned(3 downto 0);
begin
  s_logic <= std_logic_vector(s_unsig);
  process
     begin
     s_unsig <= "1110";
     wait for 1 sec;
     s_unsig <= s_unsig+1;
     wait for 1 sec;
     s_unsig <= s_unsig+1;
     wait for 1 sec;
     s_unsig <= s_unsig+1;
     wait for 1 sec;
     wait;
  end process ;
end behav ;
```



Assignments

Resizing an unsigned integer

```
signal a : unsigned(3 downto 0);
signal b : unsigned(7 downto 0);

b <= "0000" & a;
a <= b(3 downto 0);
```

- Concatenation operator: `&`
- Only in the right hand of assignments!

Alternate way

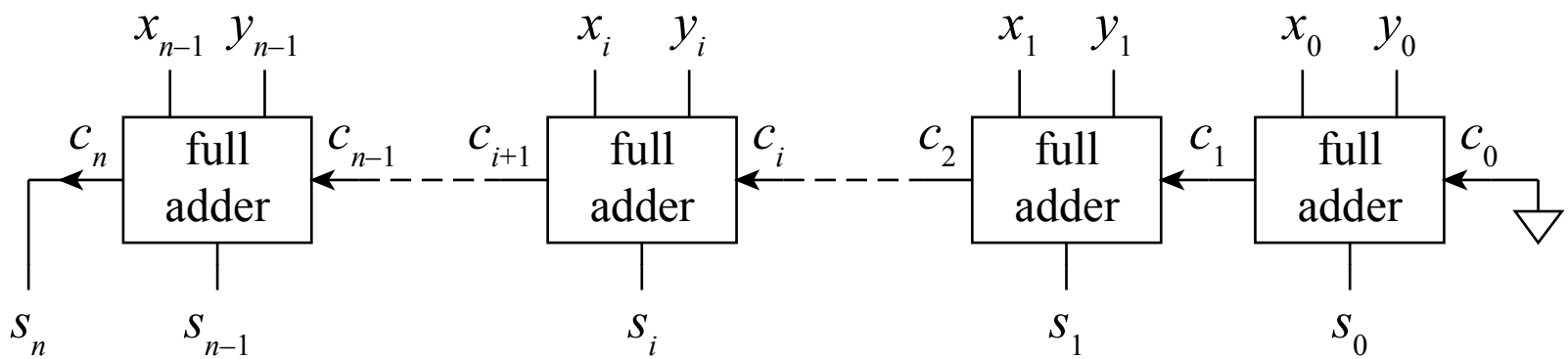
```
signal a : unsigned(3 downto 0);
signal b : unsigned(7 downto 0);

b <= resize(a,8);
a <= resize(b,4);
```

Adding Unsigned Integers

Algorithm

1	1	1	1	1	0	0	0	
	0	0	1	1	1	0	0	1
	1	1	0	0	1	1	0	0
1	0	0	0	0	0	1	0	1

Implementation

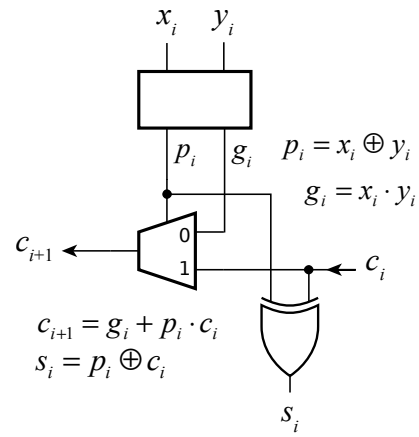


`c_out <= (a and b) or (c_in and (a xor b));` This is slooow!

Adders

Fast-carry-chain adder

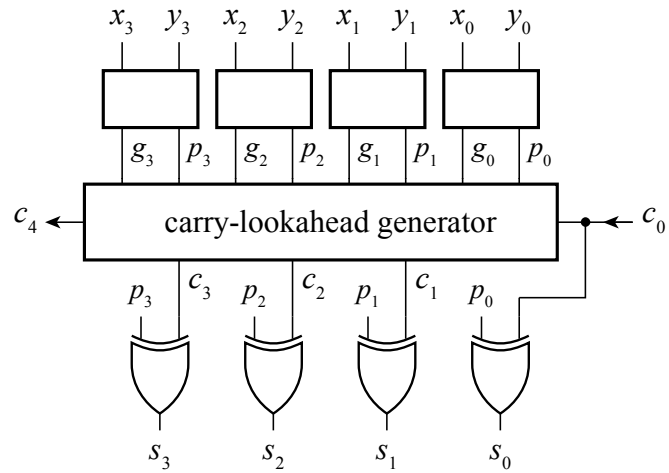
- Carry *kill*: `ki = not xi and not yi`
- Carry *propagate*: `pi = xi xor yi`
- Carry *generate*: `gi = xi and yi`




```
si = pi xor c_in;
co = gi or pi and c_in;
```

Carry-lookahed generator

- Example for 4 bits




```
c4 = g3 or (p3 and g 2)
     or (p3 and p2 and g1)
     or (p3 and p2 and p1 and g 0)
     or (p3 and p2 and p1 and p0 and c0)
```

Adder in VHDL

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
...
  signal a,b,s : unsigned(3 downto 0);
...
  s <= a + b;
```

- Do not specifiy the equations!
- The software automatically detects the best type of adder.
- This is dependent on the technology and the requirements of the designer (speed, area).

Adder with Carry Out

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
...
  signal a,b,s : unsigned(7 downto 0);
  signal temp  : unsigned(8 downto 0); --MSB:carry
  signal c_out : std_logic;
...
  temp  <= ('0' & a) + ('0' & b); -- compute sum with 9 bits
  s     <= temp(7 downto 0);      -- true sum is in LSBs
  c_out <= temp(8);               -- carry is the MSB
```

Comparing unsigned integers

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity...
  port temp   : in  unsigned(7 downto 0);
  port target : in  unsigned(7 downto 0);
  port heat   : out std_logic;
architecture...
  heat  <= '1' when temp < target - 5 else '0';
```

- Synthesis software will implement the function.

Comparing unsigned integers

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity...
  port temp   : in  unsigned(7 downto 0);
  port target : in  unsigned(7 downto 0);
  port heat   : out std_logic;
architecture...
  heat  <= '1' when temp < target - 5 else '0';
```

- Synthesis software will implement the function.