14.03.2011

©

1st midterm exam
April 20th, 2011
Wednesday 1330-1530

OVERFLOW = $C_{n+1} \oplus C_n$

Look-ahead Carry adders:

Xilinx → Fast carry logic in order simplify the look ahead carry generation circuit.

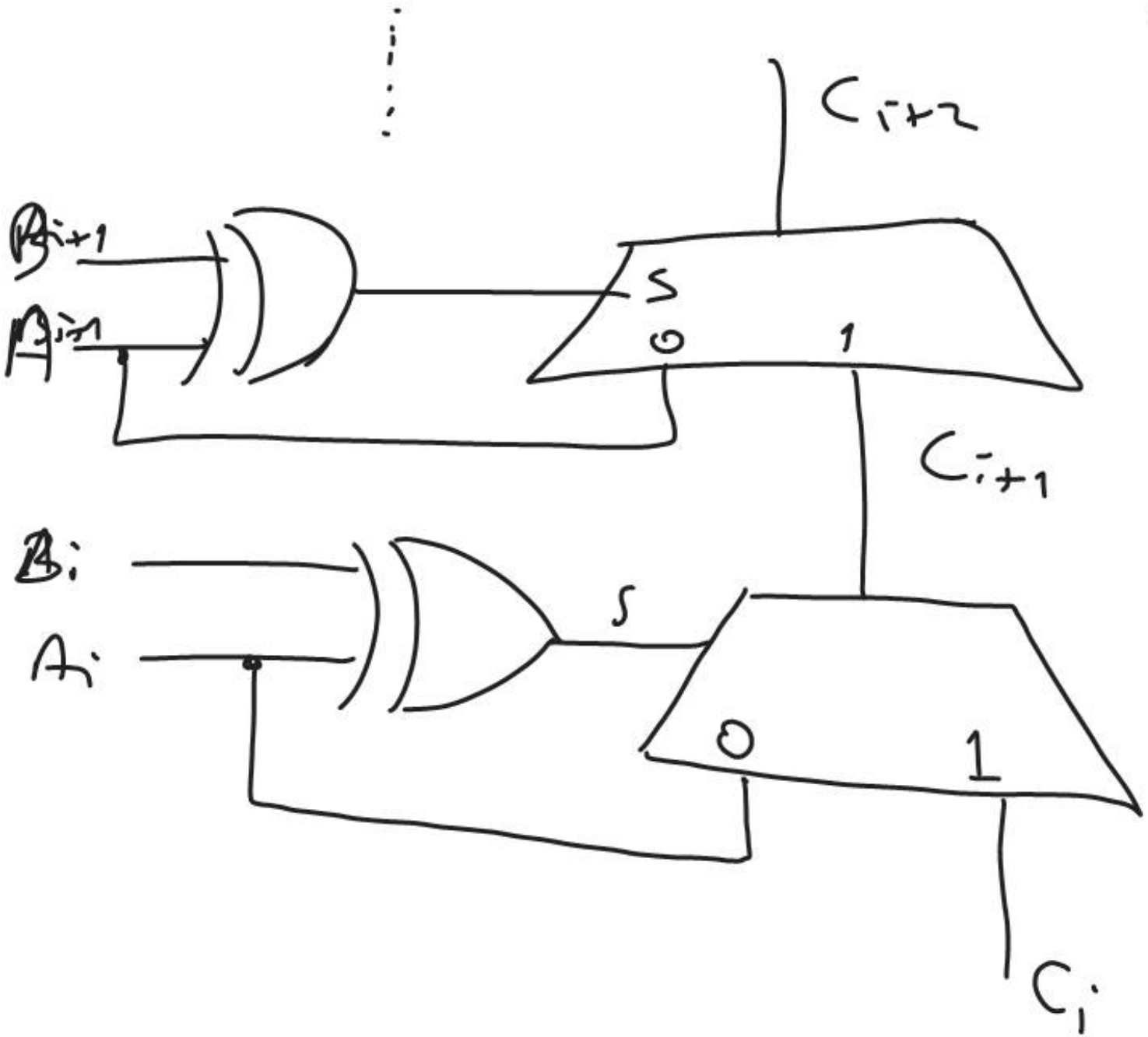
A_i	B_i	C_i	C_{i+1}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$A_i = B_i \Rightarrow C_{i+1} = A_i$

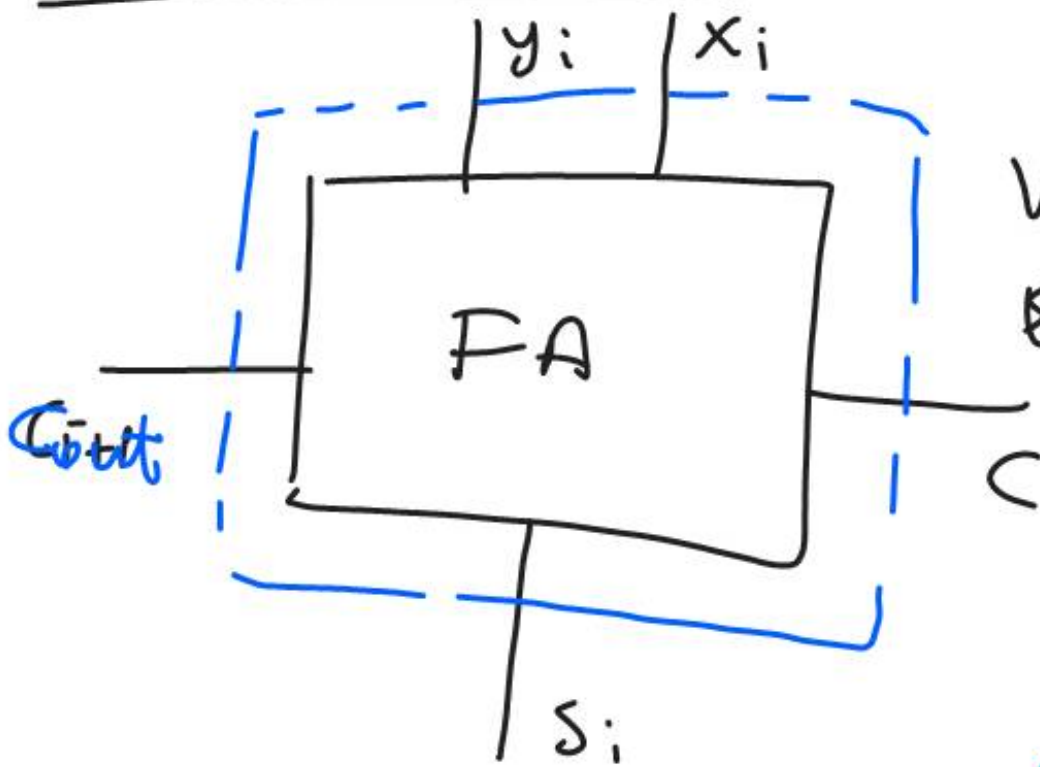
$A_i \neq B_i \Rightarrow C_{i+1} = C_i$

$A_i = B_i \Rightarrow C_{i+1} = A_i$

Fast Carry Logic



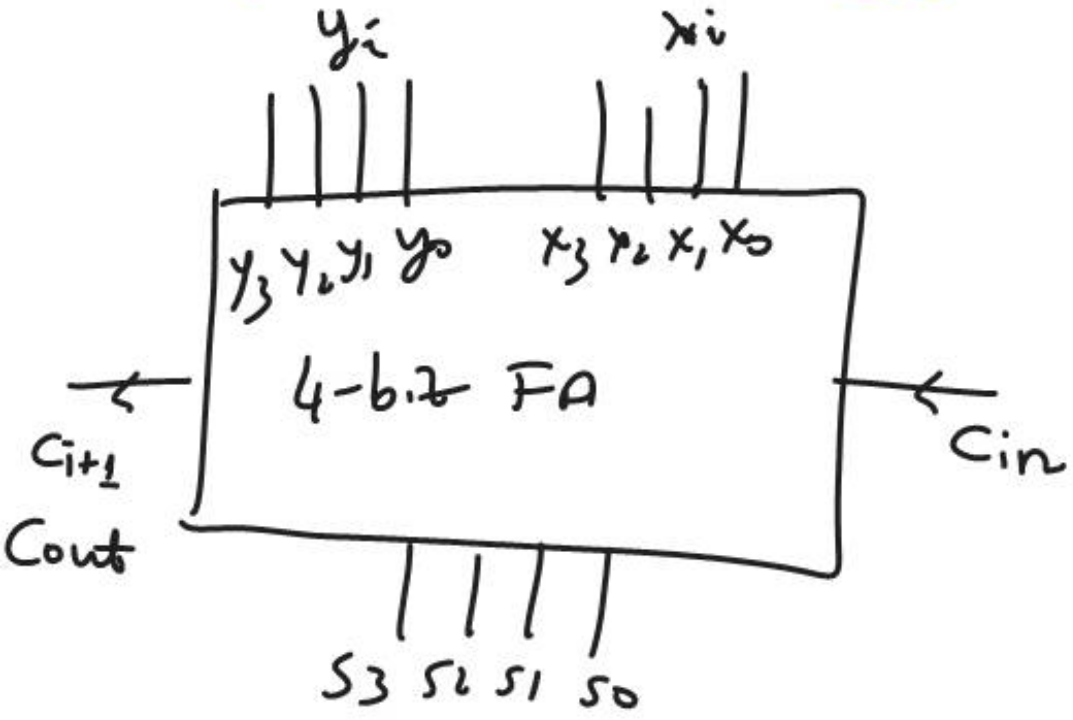
A Full-Adder (1-bit)

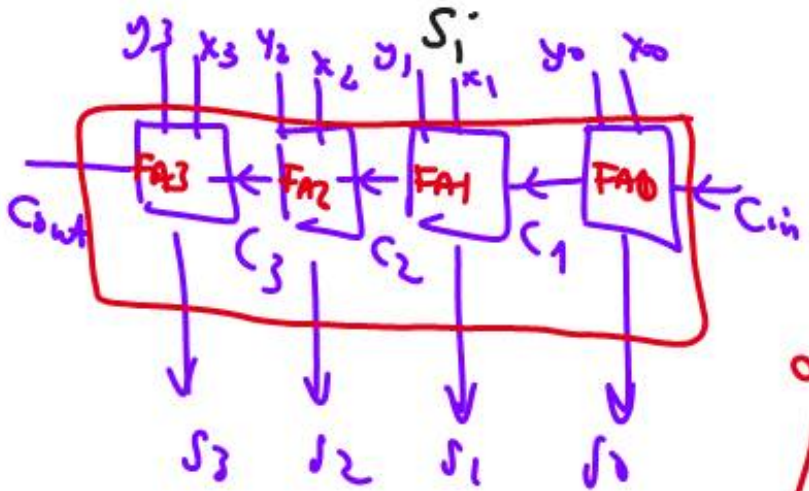


VHDL code:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY FullAdd IS
PORT( Ci, X, Y : IN STD_LOGIC;
      Cout, S : OUT STD_LOGIC);
END FullAdd;
ARCHITECTURE Structure OF FullAdd IS
BEGIN
S <= X XOR Y XOR Ci;
Cout <= (X AND Y) OR (X AND Ci) OR (Y AND Ci);
END Structure;
```

A 4-bit Full Adder






```
LIBRARY ieee;
USE ieee.____;
ENTITY Adder4 IS
PORT( X, Y : IN STD_LOGIC_VECTOR
        (3 DOWNTO 0);
  Cin : IN STD_LOGIC;
  S : OUT STD_LOGIC_VECTOR(
        3 DOWNTO 0);
  Cout : OUT STD_LOGIC);
END Adder4;
ARCHITECTURE Structure OF Adder4 IS
SIGNAL C1, C2, C3 : STD_LOGIC;
COMPONENT FullAdd
PORT( Cin, X, Y : IN STD_LOGIC;
      S, Cout : OUT STD_LOGIC);
END COMPONENT;
```

```
BEGIN
Stage 0 : FullAdd Portmap (Cin, x0, y0, S0, C1);
Stage 1 : FullAdd Portmap (C1, x1, y1, S1, C2);
Stage 2 : FullAdd PORTMAP (C2, x2, y2, S2, C3);
Stage 3 : FullAdd Portmap (C3, x3, y3, S3, Cout);
END Structure;
```

positional association

```
(Cin => C3, Cout => Cout, x => x3, y => y3, S => S3)
```

named association

If we use a package 1-bit FA:

```
LIBRARY work;
USE work.FullAdd_package.all;
LIBRARY ieee;
USE ieee.std_logic_1164.all;
PACKAGE FullAdd_package IS
 COMPONENT FullAdd
 PORT( Cin, x, y : IN STD_LOGIC;
       S, Cout : OUT STD_LOGIC);
 END COMPONENT;
END FullAdder_package;
```