Motivation

The role of Logic Synthesis in Digital Design
Motivation

Logic Design

• From a specification to an implementation
  1. Specification
  2. Representation
  3. Transformation
  4. Implementation

verification and testing play an important role
Specification

• description in words
• may include formulas (Boolean, Arithmetic)
• VHDL, Verilog
• System C
• UML
Example: full adder

• A full adder takes two inputs and a carry-in and generates the sum and carry-out
VHDL: full adder

entity adder is
  -- i0, i1 and the carry-in ci are inputs of the adder.
  -- s is the sum output, co is the carry-out.
  port (i0, i1 : in bit; ci : in bit; s : out bit; co : out bit);
end adder;

architecture rtl of adder is
begin
  -- This full-adder architecture contains two concurrent assignment.
  -- Compute the sum.
  s <= i0 xor i1 xor ci;
  -- Compute the carry.
  co <= (i0 and i1) or (i0 and ci) or (i1 and ci);
end rtl;
SystemC: full adder

FullAdder.h

SC_MODULE( FullAdder ) {
    sc_in< sc_uint<16> > A;
    sc_in< sc_uint<16> > B;
    sc_out< sc_uint<17> > result;

    void doIt( void );

    SC_CTOR( FullAdder ) {
        SC_METHOD( doIt );
        sensitive << A;
        sensitive << B;
    }
};

FullAdder.cpp

void FullAdder::doIt( void ) {
    sc_int<16> tmp_A, tmp_B;
    sc_int<17> tmp_R;

    tmp_A = (sc_int<16>) A.read();
    tmp_B = (sc_int<16>) B.read();

    tmp_R = tmp_A + tmp_B;

    result.write( (sc_uint<16>) tmp_R.range(15,0) );
}
Representation

- Truth table

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>s</th>
<th>c_{out}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Boolean expression

\[ S = a \oplus b \oplus c \]
\[ C_{out} = ab + ac + bc \]
BDDs

Motivation
Implementation

two level and-or
Implementation (cont.)

multi-level (full adder)

Cin | b | a
---|---|---
1 | 0 | 1
0 | 0 | 1
0 | 1 | 1
1 | 0 | 1
1 | 1 | 0

Slide 10
Logic Synthesis

• Transformation of Boolean functions into cost-efficient realizations.

• need:
  – suitable data structures
  – efficient algorithms