Automatic Interface Synthesis based on the Classification of Interface Protocols of IPs

ChangRyul Yun\textsuperscript{1}, DongSu Kang\textsuperscript{2}, YoungHwan Bae\textsuperscript{3}, HanJin Cho\textsuperscript{3}, KyoungSon Jhang\textsuperscript{2}

\textsuperscript{1}Agency for Defense Development, KOREA
\textsuperscript{2}Dept. of Computer Engineering, ChungNam National University, KOREA
\textsuperscript{3}Multimedia SoC Design Team, ETRI, KOREA
Outline

• Introduction & Motivation
  – Interface circuit structures

• Proposed Interface Synthesis Method
  – Interface Synthesis Flow
  – SIMPLE (Simplified Interface Protocol Description Language)
  – Classification of Interface Protocols of IPs
  – The Structures of interface Circuits
  – Interface Synthesis Algorithm

• Experiments

• Conclusion
Introduction

• IP-based Design Methodology in SoC
  – A methodology to respond to the demands of high performance, complex functionality, short time-to-market
  – One of key issues: The difficulty of IPs integration
    • IPs use various interface protocols
  – Interface circuits between IPs are necessary.

• Interface circuits
  – Time-consuming and error-prone task
  – Automatic interface circuit generation method is necessary.
Introduction

• Automatic Interface Synthesis Flow based on interface FSMs
Motivation

Flexible and highly appropriate structure depending on IP characteristics and system level requirements.

Easy and efficient interface protocol description method of IPs.

A synthesis algorithm should consider various characteristics on generation of interface circuits.

**Motivation**

**Interface circuits**

- **IP (Master)**
  - Protocol Specification
  - Interface FSM
  - Matching Information
  - User constraints

- **IP (Slave)**
  - Protocol Specification
  - Interface FSM
  - The Subset of Product FSM

A synthesis algorithm should consider various characteristics on generation of interface circuits.
Motivation

• The various characteristics of interface protocols of IPs
  – Address and data transfer characteristics
    • The number of addresses on burst
    • Shared signal or different signals for an address and data
  – No Address
  – Different clock frequencies
  – Different data widths
• These differences should be considered on an interface synthesis algorithm.
Product FSM Based Structure

• Product FSM based interface circuits
  – Consist of a product FSM and buffers
  – Allow the transfer of data without any clock cycle delay.

FSMD Based Structure

- FSMD(FSM+Data Path) based interface circuits
  - Consist of two FSMDs and buffers (Queue)
  - Every data transfer has at least two clock cycle.

The interface circuit structure based on a product FSM[6]

IPC Based Structure

- IPC (Interface Protocol Component) [10]
  - Protocol Convert
    - From transaction level to cycle level vice versa.
  - No internal buffers

Execute transaction by transaction code
Recognize transaction by signals

Proposed Interface Synthesis Flow

- Classify Interface Protocols
- Select a synthesis algorithm for interface structure depending on analysis results or user constraints
Interface FSM

• Protocol Description Methods
  – Waveform or timing diagram
  – Formal language

• SIMPLE (Proposed Description Method)
  – Simplified Interface Protocol Description Language[10]
  – Interface Protocol: Transfers + Parameters
    • Transfer: the behavior of ports on a cycle
    • Parameter
      – Address, data, transaction information and etc.
      – Used to match the different feature of IPs

Interface FSM

**SIMPLE**

Define $new_transaction, $write;  /* ⑤, ⑥ */

Interface AHB_Master {
  in  bit HREADY;  /* Ports List */
  :
  initial A;  /* ① Initial State */

  AMBA_MASTER: behavior {
    int count;  // User Defined Variable
    A: if ( $new_transaction == 1 )  /* ⑤ */  {
      HTRANS = "10";  // NONSEQ
      HADDR = $address;  /* ④ */
      HBURST = $transaction;  /* ② */
      count = $transaction_length;  /* ③ */
      if ($write == '1')  /* ⑥ */  {
        HWRITE = '1';
        if ( HREADY == '1')
          goto B;
        :
      }
      else
        goto A;
    }
  }

  A part of the description

  • Reserved Parameter
    ②: transaction type
    ③: transaction length
    ④: address

  • User Defined Parameter
    ⑤, ⑥

AHB Master interface Protocol
Interface FSM

htrans!”10”, hwrite!’1’, hready?’1’, hbust!$transaction, haddr!$address, count!$transaction_length

hready?’1’
hwdata!$wdata, count?0

htrans!”10”, hwrite!’0’, hready?’1’, hbust!$transaction, haddr!$address, hburst!$transaction
count!$transaction_length

hready?’0’
hwdata!$wdata, count?0

hready?’0’
hrdata?$rdata, count?0

htrans!”11”, hready?’1’
haddr!$address, hrdata?$rdata, count!count-1

htrans!”10”, hwrite!’1’, hready?’1’, hbust!$transaction, haddr!$address, hburst!$transaction
count!$transaction_length

τ
### The categorization of Interface protocols

#### Address & data Transfer Characteristics

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Separated ports for addresses, write data, and read data</td>
<td>AHB, OCN, VCI, OCP, PLB, etc.</td>
</tr>
<tr>
<td>B</td>
<td>Shared one port for addresses, write data, and read data</td>
<td>PCI, PCI-X, etc.</td>
</tr>
<tr>
<td>C</td>
<td>No address</td>
<td>DES, AES, Color converter, etc</td>
</tr>
</tbody>
</table>
Selection of Interface Circuit Structure

• Type I
  Product FSM-Based Structure

• Type II
  IPC-Based Structure

15
Type I

- Type I – 1

- For a pair of Category A IP
- A buffer can be employed for each port pair.
- Data can be stored or bypassed to reduce transmission delay.

- Type I – 2

- For a pair of Category A : Category B
- Generate or ignore addresses
Type I - 1

• Matching Information for a pair of PCI(M):AHB(S)

[Clock Ratio] 1 : 1 // Master IP, Slave IP
/// Buffer Size, Signal (master), MSB, LSB,
/// Signal (slave), MSB, LSB, $address or $wdata or $rdata
[Port Pair] 1 , FA , 31 , 0 , haddr , 31 , 0 , $address;
[Port Pair] 1 , FD , 31 , 0 , hwdata , 31 , 0 , $wdata;
// Burst Transaction Length, Signal(master), MSB, LSB,
// Values, Signal(slave), MSB, LSB, Values
[TM] 1 , FS , 3 , 2 , "00" , HBURST , 2 , 0 , "000";
[TM] 4 , FS , 3 , 2 , "01" , HBURST , 2 , 0 , "011";
[TM] 16 , FS , 3 , 2 , "11" , HBURST , 2 , 0 , "111";
Type - II

- IPC Based Structure
Type - II

- Type II - 4
• Matching Information for a pair of AHB(M):DES(S)

```
// Buffer Size, Master Port Name, MSB, LSB, Slave Port Name, MSB, LSB,
// Address Port, MSB, LSB, address value, $Data or $Control ;
[Port Mapping] 1, HWDATA, 31, 0, KEY, 63, 32, HADDR, 7, 0, 0x00, $Data;
[Port Mapping] 1, HWDATA, 31, 0, KEY, 31, 0, HADDR, 7, 0, 0x04, $Data;
[Port Mapping] 1, HWDATA, 31, 0, Plain, 63, 32, HADDR, 7, 0, 0x08, $Data;
[Port Mapping] 1, HWDATA, 31, 0, Plain, 31, 0, HADDR, 7, 0, 0x0C, $Data;
[Port Mapping] 1, HWDATA, 31, 0, Cipher, 63, 32, HADDR, 7, 0, 0x28, $Data;
[Port Mapping] 1, HWDATA, 31, 0, Cipher, 31, 0, HADDR, 7, 0, 0x2C, $Data;
[Port Mapping] 1, HWDATA, 0, 0, START, 0, 0, HADDR, 7, 0, 0x14, $Control;
[Port Mapping] 1, HWDATA, 0, 0, enc_dec, 0, 0, HADDR, 7, 0, 0x14, $Control;
```

- Graphical User Interface (GUI) or a script file can be used to construct the matching description.
Experiments

The comparison between manual and automatic design

<table>
<thead>
<tr>
<th>Type</th>
<th>Master : Slave</th>
<th>Area (Slices)</th>
<th>$f_{max}$</th>
<th>A / M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>M</td>
<td>A</td>
<td>M</td>
</tr>
<tr>
<td>I–1</td>
<td>AHB:OCN</td>
<td>57</td>
<td>103</td>
<td>234</td>
</tr>
<tr>
<td></td>
<td>OCN:AHB</td>
<td>112</td>
<td>374</td>
<td>197</td>
</tr>
<tr>
<td></td>
<td>BVCI:AHB</td>
<td>44</td>
<td>61</td>
<td>580</td>
</tr>
<tr>
<td></td>
<td>AHB:APB</td>
<td>66</td>
<td>121</td>
<td>268</td>
</tr>
<tr>
<td>I–2</td>
<td>AHB:PCI</td>
<td>50</td>
<td>72</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>PCI:AHB</td>
<td>82</td>
<td>212</td>
<td>258</td>
</tr>
<tr>
<td>II–1</td>
<td>AHB:PVCI (50Mhz, 32bit: 12.5Mhz, 32bit)</td>
<td>310</td>
<td>356</td>
<td>214</td>
</tr>
<tr>
<td></td>
<td>AHB:PVCI (50Mhz, 32bit: 50Mhz, 8bit)</td>
<td>244</td>
<td>316</td>
<td>225</td>
</tr>
<tr>
<td>II–3</td>
<td>AHB:PVCI (12.5Mhz, 32bit: 50Mhz, 8bit)</td>
<td>237</td>
<td>316</td>
<td>204</td>
</tr>
<tr>
<td>II–4</td>
<td>AHB:DES</td>
<td>134</td>
<td>145</td>
<td>225</td>
</tr>
</tbody>
</table>
Experiments

- Compare product FSM-based with IPC-based Structure

<table>
<thead>
<tr>
<th>Master : Slave</th>
<th>Type I (Product FSM Based)</th>
<th>Type II (IPC Based)</th>
<th>Type I / Type II</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># S</td>
<td># T</td>
<td>Area</td>
</tr>
<tr>
<td>AHB:OCN</td>
<td>9</td>
<td>53</td>
<td>203</td>
</tr>
<tr>
<td>OCN:AHB</td>
<td>12</td>
<td>89</td>
<td>374</td>
</tr>
<tr>
<td>AHB:BVCI</td>
<td>18</td>
<td>48</td>
<td>324</td>
</tr>
<tr>
<td>BVCI:AHB</td>
<td>10</td>
<td>42</td>
<td>249</td>
</tr>
<tr>
<td>AHB:PVCI</td>
<td>10</td>
<td>39</td>
<td>252</td>
</tr>
</tbody>
</table>
Experiments

- The comparison with previous work[3]

<table>
<thead>
<tr>
<th>Master:Slave</th>
<th>Clocks</th>
<th>Data Width</th>
<th>Previous Method</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># S</td>
<td># T</td>
<td></td>
<td># S</td>
</tr>
<tr>
<td>AHB:PLB</td>
<td>1:1</td>
<td>1:1</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>OCP:AHB</td>
<td>1:1</td>
<td>1:1</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>AHB:PLB</td>
<td>1:2</td>
<td>1:1</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>OCP:PLB</td>
<td>1:1</td>
<td>1:3</td>
<td>17</td>
<td>6</td>
</tr>
</tbody>
</table>

#S : The number of States, #T : The number of transitions

Conclusion

• Automatic Interface Synthesis based on protocol categorization
  – Chose an appropriate structure of interface circuits based on
    • Interface protocol category
    • Clock frequency
    • Data width
    • System level requirements (by user constraints)

  – Protocol Classification
    • Our Synthesis algorithm selects an interface circuit structure depending on the classification of interface protocols
Conclusion

• Product FSM-based structure
  – Less data transmission latency
  – Somewhat larger area and slower maximum operating frequency

• IPC-based structure
  – Less states and transitions than product FSM-based in case that IPs use different frequencies and/or data widths.

• The Performance
  – Comparable with performance of the manual designs
Thank You!