

# 반도체 IP 인터페이스의 표준화된 기술 방법

## Standardized Description Method of Semiconductor IP Interfaces

이 성 수\*★

Seongsoo Lee\*★

### Abstract

In semiconductor IP reuse, precise understanding of semiconductor IP interfaces is essential for integrated chip design. However, in general, these interfaces are described in the original designer's description style. Furthermore, their description method are not unified, so it is very difficult for the chip integration designer to understand them. This paper proposes a standardized description method of semiconductor IP interfaces. It consists of 9 items such as IP information, description level, model provision, data type, interface information, port information, signal information, protocol information, and source file. The proposed method helps the chip integration designer to understand semiconductor IP interfaces and to integrate them into a single chip.

### 요 약

반도체 IP를 재사용하여 통합 칩을 개발하기 위해서는 반도체 IP 인터페이스의 정확한 이해가 필수적이다. 그러나 이들 인터페이스는 대부분 원래 설계자의 스타일대로 기술되어 있는데다가 기술 방법이 제각각이어서 통합 칩 설계자가 이해하는데 많은 혼란이 따른다. 본 논문에서는 반도체 IP 인터페이스를 기술하는 표준화된 방법을 제안한다. 제안하는 기술 방법은 반도체 IP 인터페이스를 IP 정보, 기술 수준, 모델 제공, 데이터 타입, 인터페이스 정보, 포트 정보, 신호 정보, 프로토콜 정보, 소스 파일의 9개 항목으로 나누어 정의한다. 제안된 방법은 통합 칩 설계자가 반도체 IP의 인터페이스를 이해하고 통합 칩을 구현하는데 도움이 된다.

*Key words : standardization, description method, interface, semiconductor IP, IP reuse*

## 1. 서론

반도체 칩의 설계 시간을 단축하기 위해서 미리 설

\* School of Electronic Engineering, Soongsil University, sslee@ssu.ac.kr, 010-9182-3835

★ Corresponding author

※ Acknowledgment

“This research was supported by the Industrial Core Technology Development Program (10049095, “Development of Fusion Power Management Platforms and Solutions for Smart Connected Devices”) funded By the Ministry of Trade, industry & Energy.”

Manuscript received Aug. 29, 2014; revised Sep. 17, 2014; accepted Sep. 18, 2014

계 및 검증이 완료된 반도체 IP의 재사용이 점점 더 중요해지고 있다. 반도체 IP의 재사용은 시간과 노력을 줄이고 동작 확률과 신뢰성을 높이는 등 다양한 장점이 있어서 최근의 반도체 칩 설계에는 거의 필수적으로 쓰이고 있으나 서로 다른 설계자가 개발한 반도체 IP를 결합하여 시뮬레이션 및 검증을 수행하는데 어려움을 겪고 있다.

반도체 IP의 재사용에서 가장 어려운 점은 서로 다른 방식으로 개발된 인터페이스를 통합 칩 설계자가 이해하기가 어렵다는 점이다. 반도체 IP의 상당수는 지적재산권을 보호하기 위해서 내부 구조를 전혀 보여주지 않고 최종적인 레이아웃 또는 넷리스트만을 제공하기 때문에 인터페이스의 정확한 이해 없이는 반도체 IP의 통합이 매우 어렵지만, 이들 인터페이스는 대부분 원래 설계자의 이해 수준에 맞추어져 기술

Table 1. Description method of semiconductor IP interfaces.

표 1. 반도체 IP 인터페이스의 기술 방법.

항목	하위 항목	항목 값의 형태	선택값
IP 정보	IP 이름	사용자값	-
	IP 기능	사용자값	-
	IP 설명	사용자값	-
기술 수준		선택값	port_level 또는 signal_level
모델 제공	실행 가능 동작 모델	선택값	Y 또는 N
	구현 가능 HDL 모델	선택값	Y 또는 N
데이터 타입	데이터 타입 이름	사용자값	-
	데이터 형태	선택값	boolean, std_logic, char, int, double, 또는 enumeration
	데이터 값의 범위	사용자값	-
	데이터 길이	사용자값	-
인터페이스 정보	다이어그램	사용자값	-
	인터페이스 이름	사용자값	-
	입출력 포트	사용자값	-
	동작 설명	사용자값	-
포트 정보	다이어그램	사용자값	-
	입출력 포트 이름	사용자값	-
	데이터/제어 신호	사용자값	-
	데이터 타입	사용자값	-
	기능 타입	선택값	transRead, transWrite, messSense, 또는 messEmit
	파일 이름	사용자값	-
신호 정보	데이터/제어 신호	사용자값	-
	데이터 타입	사용자값	-
	기능 타입	선택값	transRead, transWrite, messSense, messEmit, 또는 transState
	파일 이름	사용자값	-
프로토콜 정보		사용자값	-
소스 파일		사용자값	-



Fig. 1. Representation of data flows and control flows.

그림 1. 데이터 흐름 및 제어 흐름의 표기 방법.

(description)되어 있는데다가, 기술하는 방법도 제각각이어서 이들 반도체 IP를 설계하지 않은 통합 칩 설계자가 이해하는데 많은 혼란이 따른다. 더욱이 어떤 반도체 IP는 인터페이스에 사용되는 포트만 기술되고 어떤 반도체 IP는 개별 신호와 동작까지 기술되기 때문에 이들 반도체 IP를 연결할 때 각 반도체 IP가 기술된 수준의 차이로 인해 이들 반도체 IP를 통합하는데 큰 어려움을 겪고 있다.

이러한 문제점을 해결하는 방법의 하나는 반도체 IP 인터페이스를 기술하는 방법을 표준화하는 것이다. 그러나 표준화가 지나치게 세밀하면 설계자의 자유를 침해하여 설계 능력이 떨어질 수 있고, 표준화가 지나치게 간략하면 너무 많은 항목이 제각각으로 기술되어 다른 사람이 이해하기 어려울 수 있다.[1][2]

본 논문에서는 이러한 문제점을 해결하기 위해 반

도체 IP 인터페이스를 기술하는 표준화된 방법을 제안한다. 이렇게 표준화된 반도체 IP 인터페이스의 기술 방법을 통하여 통합 칩 설계자가 각 반도체 IP의 인터페이스를 쉽게 이해하고 각 반도체 IP를 쉽게 통합하여 통합 칩을 설계할 수 있다.

## II. 반도체 IP 인터페이스의 표준화된 기술 방법

반도체 IP 인터페이스의 표준화된 기술 방법은 표 1과 같이 IP 정보, 기술 수준, 모델 제공, 데이터 타입, 인터페이스 정보, 포트 정보, 신호 정보, 프로토콜 정보, 소스 파일의 9개 항목으로 정의된다. 9개 항목 및 이들의 하위 항목은 사용자값 또는 선택값 중 하나의 형태를 가지며, 선택값은 표 1에 지정된 몇 개

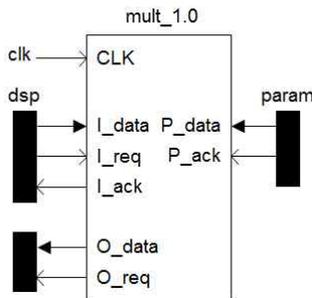


Fig. 2. Example of semiconductor IP interface.  
그림 2. 반도체 IP 인터페이스의 예.

Table 2. Example of IP information.  
표 2. IP 정보의 예.

IP 이름	mult_1.0
IP 기능	$O = P * I$
IP 설명	이 IP는 곱셈에 필요한 파라미터인 P, I를 입력받아 곱셈 결과인 O를 출력한다.

Table 3. Example of description level.  
표 3. 기술 수준의 예.

기술 수준	signal_level
-------	--------------

Table 4. Example of model provision.  
표 4. 모델 제공의 예.

실행 가능 동작 모델	Y
구현 가능 HDL 모델	N

Table 5. Example of data type.  
표 5. 데이터 타입의 예.

데이터 타입 이름	double_data
데이터 형태	double
데이터 값의 범위	$-1.79e308 \sim 1.79e308$
데이터 길이	8바이트
데이터 타입 이름	bool_ctrl
데이터 형태	boolean
데이터 값의 범위	true, false
데이터 길이	1비트
데이터 타입 이름	std_logic
데이터 형태	std_logic
데이터 값의 범위	0, 1, U, X, Z, W, L, H, -
데이터 길이	1비트

의 값 중에서 하나를 선택하여 사용한다. 인터페이스 정보 및 포트 정보의 다이어그램에서 데이터 흐름 및 제어 흐름은 그림 1의 표기법을 사용한다. 이 기술 방법은 한국정보통신기술협회의 표준[3] 및 VSI

Alliance의 표준[4]을 부분적으로 참고하였다.

### 1. IP 정보

IP 정보는 인터페이스가 적용되는 반도체 IP의 이름, 기능 및 일반적인 설명을 나타낸다. 표 2는 그림 2와 같은 반도체 IP 인터페이스에서 IP 정보를 나타낸 예인데, mult\_1.0이라는 이름의 반도체 IP가 곱셈에 필요한 파라미터인 P, I를 입력받아 곱셈 결과인 O를 출력하는 기능을 수행한다는 것을 나타낸다.

### 2. 기술 수준

기술 수준은 인터페이스가 어느 수준까지 기술되었는지를 나타내며, port\_level인 경우에는 포트 수준만, signal\_level인 경우에는 포트 수준과 신호 수준 모두 기술되었음을 의미한다. 표 3은 그림 2와 같은 반도체 IP 인터페이스에서 기술 수준을 나타낸 예인데, 이 반도체 IP는 포트 수준과 신호 수준 모두 기술되어 있다는 것을 나타낸다.

### 3. 모델 제공

모델 제공은 인터페이스가 실행 가능한 동작 모델 및 구현 가능한 HDL 모델을 제공하는지의 여부를 나타낸다. 표 4는 그림 2와 같은 반도체 IP 인터페이스에서 모델 제공을 나타낸 예인데, 이 반도체 IP는 시뮬레이션 실행이 가능한 C/C++ 또는 HDL 파일은 제공되지만, 하드웨어 구현이 가능한 HDL 파일은 제공되지 않는다는 것을 나타낸다.

### 4. 데이터 타입

데이터 타입은 인터페이스에서 사용하는 데이터의 타입을 나타낸다. 이 항목은 데이터 타입 이름, 데이터 형태, 데이터 값의 범위, 데이터 길이를 표기하여야 하며, 여러 가지의 데이터 타입이 사용되는 경우 각 데이터 타입마다 따로따로 표기하여야 한다. 데이터 타입은 char, std\_logic과 같이 C/C++이나 HDL에서 표준으로 사용하는 타입일 수도 있고, IP 인터페이스 개발자가 정의한 타입일 수도 있다. 데이터 길이는 C/C++ 변수 또는 HDL 신호의 비트수(또는 바이트 수)로 표기된다. 표 5는 그림 2와 같은 반도체 IP 인터페이스에서 데이터 타입을 나타낸 예인데, 이 반도체 IP는 배정도 (double-precision) 실수, 부울 (Boolean) 변수, 표준 로직 변수의 데이터 타입을 사용한다는 것을 나타낸다.

### 5. 인터페이스 정보

인터페이스 정보는 인터페이스에서 사용하는 물리적/논리적 인터페이스를 나타낸다. 다이어그램은 인터페이스에서 데이터 및 제어 신호의 방향, 연결 관계 및 블로킹 특성을 나타낸다. 하나의 인터페이스가 다수의 포트로 구성되는 경우에는 각각의 포트마다 하나씩만 표기하며, 데이터 신호의 방향과 제어 신호의 방향을 모두 표기한다. 하나의 포트가 다수의 제어 신호를 사용하는 경우에는 주된 제어 신호의 방향에 대해서만 표기한다. 표 6은 그림 2와 같은 반도체 IP 인터페이스에서 인터페이스 정보를 나타낸 예인데, 이 반도체 IP는 다이어그램과 같이 dsp, parm이라는 두 개의 인터페이스를 사용하며, dsp라는 인터페이스를 통해 파라미터인 I를 입력하고 수행 결과인 O를 출력하며 parm이라는 인터페이스를 통해 파라미터인 P를 입력한다는 것을 나타낸다.

표 6의 다이어그램에서 DSP와 I 사이의 포트는 그림 2에서와 같이 두 개의 제어 신호인 Lreq와 Lack를 사용하지만 주된 제어 신호는 Lack이기 때문에 이 제어 신호의 방향인 ←만을 표기하고, Lreq 제어 신호의 방향인 →는 표기하지 않는다.

### 6. 포트 정보

포트 정보는 인터페이스에서 사용하는 입출력 포트를 나타낸다. 이 항목은 인터페이스에서 사용하는 모든 입출력 포트 및 데이터/제어 신호를 나타내어야 하나, 기술 수준이 port\_level인 경우에는 신호 정보가 나타나지 않으므로 데이터/제어 신호는 생략된다. 다이어그램은 포트에서 데이터 및 제어 신호의 방향, 연결 관계 및 블로킹 특성을 나타낸다. 기술 수준이 port\_level인 경우에는 신호 정보가 나타나지 않으므로 다이어그램은 생략된다. 파일 이름은 각 포트의 동작이나 구현이 기술된 C/C++ 또는 HDL 파일의 이름을 의미한다.

기능 타입은 transRead, transWrite, messSense, messEmit 중 하나여야 하며, 그 의미는 다음과 같다. 여기에서 blocking\_description은 blocking 또는 non-blocking 중 하나여야 하며, 해당 포트가 블로킹 또는 논블로킹인 것을 의미한다.

- transRead(blocking\_description): 해당 포트에서 데이터 값을 읽어들이.
- transWrite(blocking\_description): 해당 포트에 데이터 값을 써넣음.
- messSense(blocking\_description): 해당 포트에 진

Table 6. Example of interface information.

표 6. 인터페이스 정보의 예.

다이어그램		
	인터페이스 이름	dsp
입출력 포트	I, O, C	
동작 설명	곱셈에 필요한 파라미터인 I를 입력하고 곱셈을 수행한 결과인 O를 출력함	
인터페이스 이름	param	
입출력 포트	P	
동작 설명	곱셈에 필요한 파라미터인 P를 입력함	

달되는 이벤트를 인식함.

- messEmit(blocking\_description): 해당 포트를 통해 이벤트를 발생함.

시스템 클럭이나 리셋 신호 등 다수의 포트에 공통으로 사용되는 데이터/제어 신호도 각 포트마다 모두 명시해야 한다. 각각의 포트는 다수의 데이터/제어 신호로 구성되기 때문에, 포트의 데이터 타입과 기능 타입은 해당 포트가 데이터 신호가 있는 경우에는 주된 역할을 수행하는 데이터 신호의 데이터 타입과 기능 타입을 따르며 제어 신호만 있는 경우에는 주된 역할을 수행하는 제어 신호의 데이터 타입과 기능 타입을 따른다.

표 7은 그림 2와 같은 반도체 IP 인터페이스에서 포트 정보를 나타낸 예인데, 이 반도체 IP는 다이어그램과 같이 CLK, I, O, P라는 네 개의 포트를 사용한다. CLK 포트는 논블로킹으로 전달되는 제어 신호인 clk를 사용하며, clk 신호로 전달되는 이벤트를 논블로킹으로 인식한다. I 포트는 블로킹으로 전달되는 데이터 신호인 I\_data, 논블로킹으로 전달되는 제어 신호인 Lreq, Lack를 사용하며, I\_data 신호로 전달되는 데이터 값을 블로킹으로 읽어들이고, O 포트는 논블로킹으로 전달하는 데이터 신호인 O\_data와 논블로킹으로 전달하는 제어 신호인 O\_req를 사용하며, O\_data 신호로 전달하는 데이터 값을 논블로킹으로 써넣는다. P 포트는 논블로킹으로 전달되는 데이터 신호인 P\_data, 논블로킹으로 전달되는 제어 신호인 P\_ack를 사용하며, P\_data 신호로 전달되는 데이터 값을 논블로킹으로 읽어들이고.

7. 신호 정보

신호 정보는 인터페이스에서 사용하는 데이터/제어 신호를 나타낸다. 이 항목은 인터페이스에서 사용하는 모든 데이터/제어 신호를 나타내어야 하나, 기술 수준이 port\_level인 경우에는 신호 정보가 나타나지 않으므로 이 항목은 생략된다. 파일 이름은 각 데이터/제어 신호의 동작이나 구현이 기술된 C/C++ 또는 HDL 파일의 이름을 의미한다.

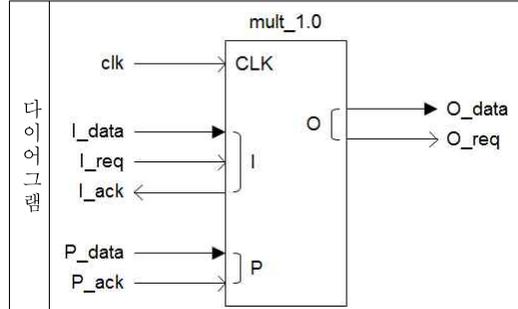
기능 타입은 transRead, transWrite, messSense, messEmit, transState 중 하나여야 하며, 그 의미는 다음과 같다. 여기에서 모든 variable과 state는 반도체 IP 내부의 variable 이름과 state 이름을 의미한다.

- transRead(variable): 해당 데이터 신호의 값을 지정된 variable에 읽어들이.
- transWrite(variable or constant): 지정된 variable 또는 constant의 값을 해당 데이터 신호에 씌넣음.
- messSense(variable or constant): 해당 제어 신호로 전달되는 이벤트를 인식함. 이때 해당 제어 신호가 지정된 variable 또는 constant의 값을 가질 때 이벤트를 인식함.
- messEmit(variable or constant): 해당 제어 신호를 통해 이벤트를 발생함. 이때 해당 제어 신호는 지정된 variable 또는 constant의 값을 가짐.
- transState(variable or constant, stateA = StateDescriptionA, {stateB = StateDescriptionB, ...}): 해당 제어 신호를 통해 반도체 IP 내부에 있는 FSM의 stateA, stateB, ...를 각각 StateDescriptionA, StateDescriptionB, ...와 같이 변경함. 이때 해당 제어 신호가 지정된 variable 또는 constant의 값을 가질 때 이벤트를 인식하며, stateA, stateB, ...가 변경되는 값인 StateDescriptionA, StateDescriptionB, ...는 variable 또는 constant로 지정할 수도 있고, C/C++언어 등으로 기술된 알고리즘으로 지정할 수도 있음.

표 8은 그림 2와 같은 반도체 IP 인터페이스에서 신호 정보를 나타낸 예인데, 이 반도체 IP는 clk, I\_data, I\_req, I\_ack, O\_data, O\_req, P\_data, P\_ack라는 여덟 개의 신호를 사용한다. 신호 정보는 포트 정보와 그 표기법이 유사하나, 포트 정보에서는 하나의 포트에서 주된 신호에 대해서만 데이터 타입과 기능 타입을 나타내는 반면에 신호 정보에서는 모든 신호에 대해서 데이터 타입과 기능 타입을 나타낸다. 또한 포트 정보에서는 각 포트의 블로킹 여부만 나타나, 신호 정보에서는 각 신호가 반도체 IP 내부 또는 외부로 어떤 변수나 값을 주고받는지 나타낸다. 예

Table 7. Example of port information.

표 7. 포트 정보의 예.



입출력 포트 이름	CLK
데이터/제어 신호	clk
데이터 타입	bool_ctrl
기능 타입	messSense(Non_Blocking)
파일 이름	clk.cpp

입출력 포트 이름	I
데이터/제어 신호	I_data, I_req, I_ack
데이터 타입	double_data
기능 타입	transRead(Blocking)
파일 이름	I.cpp

입출력 포트 이름	O
데이터/제어 신호	O_data, O_req
데이터 타입	double_data
기능 타입	transWrite(Non_Blocking)
파일 이름	O.cpp

입출력 포트 이름	P
데이터/제어 신호	P_data, P_ack
데이터 타입	double_data
기능 타입	transRead(Non_Blocking)
파일 이름	P.cpp

를 들어 I\_data의 기능 타입은 transRead(MULTIN1)인데 이는 I\_data를 통해 반도체 IP 외부 신호로부터 내부의 MULTIN1이라는 신호에 읽어들이는 것을 의미한다. O\_data의 기능 타입은 transWrite(MULTOUT)인데 이는 반도체 IP 내부의 MULTOUT이라는 신호를 O\_data를 통해 외부 신호에 씌넣는다는 것을 의미한다. I\_req의 기능 타입은 messEmit(0 or 1)인데 이는 I\_req로 0 또는 1의 값을 외부로 발신하여 전달한다는 것을 의미한다. P\_ack의 기능 타입은 messSense(1)인데 이는 P\_ack로 외부에서 전달된 값이 1일 때를 감지한다는 것을 의미한다.

Table 8. Example of signal information.

표 8. 신호 정보의 예.

데이터/제어 신호	clk
데이터 타입	bool_ctrl
기능 타입	messSense(CLK)
파일 이름	clk.cpp
데이터/제어 신호	I_data
데이터 타입	double_data
기능 타입	transRead(MULTIN1)
파일 이름	I.cpp
데이터/제어 신호	L_req
데이터 타입	bool_ctrl
기능 타입	messEmit(0 or 1)
파일 이름	L.cpp
데이터/제어 신호	L_ack
데이터 타입	bool_ctrl
기능 타입	messSense(1)
파일 이름	L.cpp
데이터/제어 신호	O_data
데이터 타입	double_data
기능 타입	transWrite(MULTOUT)
파일 이름	O.cpp
데이터/제어 신호	O_req
데이터 타입	bool_ctrl
기능 타입	messEmit(1)
파일 이름	O.cpp
데이터/제어 신호	P_data
데이터 타입	double_data
기능 타입	transRead(MULTIN2)
파일 이름	P.cpp
데이터/제어 신호	P_ack
데이터 타입	bool_ctrl
기능 타입	messSense(1)
파일 이름	P.cpp

8. 프로토콜 정보

프로토콜 정보는 인터페이스에서 사용하는 포트 및 신호에 적용되는 프로토콜을 나타낸다. 이 항목에서는 기술 방법을 한정하지 않으며 C/C++, HDL 이외에도 기존의 다양한 기술 방법을 적용할 수 있으나, 포트 정보와 신호 정보에서 사용한 기술 방법을 표 9와 같이 결합하여 사용하는 것이 편리하다.

표 9는 그림 2와 같은 반도체 IP 인터페이스에서 프로토콜 정보를 나타낸 예인데, 이 반도체 IP에서 I, O, P 포트의 동작은 다음과 같다. I 포트는 블로킹으로 데이터를 읽어들이는 포트로서, 초기화된 후에 먼

Table 9. Example of protocol information.

표 9. 프로토콜 정보의 예.

프로토콜 정보	<pre> I: transRead(Blocking) {   L_req: messEmit(1);   again:   IF (L_ack: messSense(1))     L_req: messEmit(0);     L_data: transRead(MULTIN1);   ELSE     L_req: messEmit(1);     GOTO(again); }  O: transWrite(Non-Blocking) {   O_ack: messEmit(1);   O_data: transWrite(MULTOUT); }  P: transRead(Non-Blocking) {   IF (P_ack: messSense(1))     P_data: transRead(MULTIN2); }                 </pre>
---------	--

저 L\_req 신호를 1로 만든다. 이후, L\_ack 신호가 1이면 L\_req 신호를 0으로 만들고 L\_data 신호를 반도체 IP 내부의 MULTIN1 신호에 읽어들이고, L\_ack 신호가 0이면 L\_req를 1로 만들고 L\_ack 신호가 1이 될 때까지 기다린다. O 포트는 논블로킹으로 데이터를 내보내는 포트로서, O\_ack 신호를 1로 만들고 반도체 IP 내부의 MULTOUT 신호를 O\_data로 내보낸다. P 포트는 논블로킹으로 데이터를 읽어들이는 포트로서, P\_ack 신호가 1이면 P\_data 신호를 반도체 IP 내부의 MULTIN2 신호에 읽어들인다.

9. 소스 파일

소스 파일은 인터페이스에서 사용하는 포트 및 신호의 동작을 기술한 소스 파일을 나타낸다. 이 항목은 타 항목과는 달리 표준 문서가 아닌 파일로 제공되며, 파일을 제공할 필요가 없는 경우에는 생략할 수 있다. 이 항목에서는 기술 방법을 한정하지 않으며, C/C++, HDL 이외에도 기존의 다양한 기술 방법을 적용할 수 있다.

III. 결론

반도체 IP를 재사용하여 통합 칩을 설계하기 위해서는 반도체 IP가 가지고 있는 인터페이스의 구조 및 동작을 올바르게 이해하는 것이 필수적인데, 기존에는 이러한 반도체 IP 인터페이스를 기술하는 방법이

설계자에 따라 제각각이어서 통합 칩 설계자에게 큰 부담이 되어왔다. 본 논문에서는 이러한 문제점을 해결하기 위해 반도체 IP 인터페이스를 기술하는 표준화된 방법을 제안하였다. 제안된 기술 방법은 반도체 IP 인터페이스를 IP 정보, 기술 수준, 모델 제공, 데이터 타입, 인터페이스 정보, 포트 정보, 신호 정보, 프로토콜 정보, 소스 파일의 9개 항목으로 나누어 정의하였다. 제안된 기술 방법을 IDEC에서 제공한 설계 도구를 사용하여 다수의 반도체 IP 인터페이스에 적용해본 결과, 이들 반도체 IP 인터페이스를 간단하고 명확하게 기술할 수 있었으며, 거의 모든 반도체 IP 인터페이스를 효과적으로 기술할 수 있을 것으로 판단된다. 본 논문에서 설명한 반도체 IP 인터페이스의 기술 방법은 향후 반도체 IP의 재사용 및 통합 칩 설계에 큰 도움이 될 것으로 생각된다.

## References

- [1] S. Lee, "Standardized Description Method of Image Acquisition Characteristics Tests for Image Sensors and Modules, Journal of IKEEE, vol. 18, no. 1, pp. 64-76, Mar. 2014.
- [2] S. Lee, "Standard Image Acquisition Characteristics Tests for Image Sensors and Modules", Journal of IKEEE, vol. 18, no. 1, pp. 77-95, Mar. 2014.
- [3] Telecommunications Technology Association, TTAK.OT-10.0289, "Description and Architecture of Semiconductor IP Interface", 2010.
- [4] VSI Alliance, "System-Level Interface Behavioral Documentation Standard (SLD) 1 1.0", 2000.

## BIOGRAPHY

### Seongsso Lee (Life Member)



1991 : BS degree in Electronic Engineering, Seoul National University.

1993 : MS degree in Electronic Engineering, Seoul National University.

1998 : PhD degree in Electrical Engineering, Seoul National University.

1998~2000 : Research Associate, University of Tokyo

2000~2002 : Research Professor, Ewha Womans University

2002~Now : Associate Professor in School of Electronic Engineering, Soongsil University

<Main Interest> HEVC, Low-Power SoC Design, Multimedia SoC Design, Battery Management