



김진태 | 센터장 |  
전자부품연구원 IP/SoC 지원센터  
E-Mail : jtkim@keti.re.kr

# IP 전달물 규정

## IP Deliverables Specifications

### 서론

다기능, 고성능, 고품질의 차별화 된 제품을 요구하는 소비자의 욕구에 부응하기 위하여 반도체 제품이 각종 제품에 널리 사용되고 있다. 반도체 제품을 만들기 위해서는 반도체 설계기술과 제조기술을 동시에 보유하고 있어야 하는데, 우리나라의 경우 반도체 제조기술은 우수한 반면 설계기술과 설계인력은 부족한 실정이며, 메모리 분야에서는 세계적인 경쟁력을 갖추고 있으면서도 비메모리 반도체 분야에서는 경쟁력이 매우 낮은 형편이다.

이러한 우리나라의 특수한 상황은 젓혀놓고 보아도 전 세계 반도체 업계는 비메모리 반도체 분야의 설계부문에서 커다란 난관에 봉착하고 있다. 즉, 날로 높아지는 소비자의 요구를 충족시킬 수 있는 반도체 제조기술을 가지고 있음에도 불구하고(과거에는 반도체 제조기술의 낙후로 비메모리 반도체의 제조가 불가능했으나 최근에는 반도체 설계기술의 낙후로 제조기술을 따라가지 못하고 있음) 소비자가 요구하는 제품의 핵심인 비메모리 반도체 제품을 설계하는데 많은 시간이 소요되어 특단의 대책이 강구되어야 하는 상황이 되었다. 이러한 문제를 해결하고자 전 세계 반도체 업계는 지금까지 비메모리 반도체 설계/개발을 통해 축적한 설계 데이터를 재활용(design and reuse)하여 비메모리 반도체의 설계시간을 단축하고자 하는 목표를 두고 1990년대 중반부터 IP(Intellectual Property)라는 개념을 도입하여 분주하게 움직이고 있다. 이러한 움직임 가운데는 기존의 설계데이터를 재활용하기 위하여 설계데이터를 표준화하여 유통시킨다는 개념(VSIA의 설립 목표)이 들어 있고, 설계데이터의 표준화 및 유통과 관련하여 각종 단체(Design & Reuse, the VCX Software 등)들이 생겨나고 있다. 국내에서도 이러한 움직임에 편승하여 여러 IP 관련 과제/기관들이 생성/설립되고 있으며, 정부주도의 대형 국책기술개발사업인 차세대성장동력산업에서도 차세대 SoC 반도체 산업이 매우 중요한 부분을 차지하게 되었다.

본 내용은 이러한 비메모리 산업의 발전을 위해 탄생한 IP를 활용함에 있어 IP 사용자가 편리하게 IP를 사용할 수 있도록 어떤 형태(유/무형)의 전달물들이 필요하며 이를 위해 IP 개발자들이 어떠한 형태의 전달물을 제공해야 하는가에 대한 내용이다.

전체적인 내용은 VSIA의 표준안을 근거로 전달물에 대한 규정과 각 IP의 종류에 따른 필요사항들에 대해 내용을 전개하였으며, IP의 전달물(Deliverables)에 대해 알아보기 전에 내용의 이해를 돕고자 IP가 무엇인가에 대해 간략하게 정리하고 본 논고의 중심이 되는 전달물(Deliverables)에 대해 논의하였다.

### IP의 의미와 종류

#### 1. IP

IP(Intellectual Property)의 실질적인 의미는 특허, 저작권 및 거래에 의해 보호되는 상품이나 기술, 소프트웨어 등을 가리키는 표현으로 우리가 일 상에서 말하는 특허가 그 대표적인 주자라 할 수 있다. 그러나 현재 전기전자 반도체분야에서 IP라고 부르는 주체는 실제의미의 IP가 아닌 전기전자 산업에서 활용되는 반도체 기술을 표현하는 단어로 축소되어 나타난다. 이러한 상황에서 국내에서는 IP의 표현을 많이 사용하고 있으나 이러한 IP와 관련하여 범국가적인 표준을 제정하고자 반도체 회사 및 다국적 EDA 기관들이 참여하고 있는 VSIA(Virtual Socket Interface Alliance)에서는 전기전자 반도체분야의 설계자로 재활용 기술에 대해 VC(Virtual Component)라는 표현을 주로 사용하고 있으며, IP라는 표현에 대해서는 상기에 언급한 바와 같이 좀더 광의적인 의미로 해석하고 있다.

#### 2. IP 구분

	Design Flow	Representation	Libraries	Technology	Protability
<b>Soft</b> Not Predictable Very Flexible	System Design RTL Design	Behavioral RTL	N/A	Technology Independent	Unlimited
<b>Firm</b> Predictable Flexible	Floor Planning Synthesis Placement	RTL & Blocks Netlist	Reference Library . Footprint . Timing model . Wiring model	Technology Generic	Library Mapping
<b>Hard</b> Very Predictable Not Flexible	Routing Verification	Polygon Data	Process specific Library & design Rules . Characterized Cells . Process rules	Technology Fixed	Process Mapping

[그림 1] VSIA IP 분류기준

상기의 [그림 1]은 현재 VSIA에서 규정하고 있는 IP의 분류기준이다. VSIA에서는 IP를 3가지로 분류하고 있으며, 이러한 분류기준은 IP의 형태 및 Flexibility, Predictability 등을 기준으로 구분하고 있다. 최근에는 이러한



분류기준에 추가하여 각종 알고리즘 수준의 개발환경이 등장하고 알고리즘 수준의 설계데이터로부터 하드웨어 설계가 가능해졌으며, 알고리즘 수준의 기본 설계업무가 중요시 되면서 Model IP에 대한 부가적인 중요도가 높아지고 있다. 이러한 Model IP의 전달물에 대해서는 VSI에 명시되어 있지 않기 때문에 VSI 기준의 IP 전달물 규정에 대해 논의한 후 마지막 부분에 Model IP의 특성 및 전달물에 대해 알아보기로 하겠다.

## IP Deliverables

IP 전달물이라 하는 것은 많은 의미를 내포한다. 특히 IP에 있어 전달물에 대한 규정은 VSIA의 Documents나 Standard 등에 나타나 있듯이 매우 방대하며, 이를 모두 적용하여 IP를 개발한다는 것은 실질적으로 불가능하다고 할 수도 있을 것이다. 그러나 IP를 개발하고 이를 재활용하기 위해서는 IP 활용을 위한 기본적인 데이터 문서에서 시작하여 원시 코드(source code : HDL, GDS 등) 등의 자료가 분명히 필요하며, 이를 보유하고 있지 못할 경우 IP로서의 가치를 상실하게 된다.

VSIA에서 규정하고 있는 각 IP 분류별 전달물 규정은 다음의 표와 같다.

절	Deliverable	현재 사용되는 형식	Soft VC	Soft VC	Soft VC	비고
3.1	<b>사용자 지침</b>					
	사양	Document	M	M	M	
	VC정보	Document	M	M	M	
	VC정보의 검증	Document	M	M	M	
	버전 정보	Document	M	M	M	
	버그 정보	Document	M	M	M	
	어플리케이션 노트	Document	R	R	R	
3.2	<b>시스템 구조</b>					
3.2.1	시스템검증모델	C,C++,VHDL,Verilog	CR	CR	CR	
3.3	<b>시스템 설계정보</b>					
3.3.1	검증용테스트벤치	VHDL,Verilog,+PLI	R	R	R	
3.3.2	동작 모델	C/C++,VHDL,Verilog	R	CM	M	OK to use RTL, if available
3.3.3	프로세서 모델	C/C++,VHDL,Verilog	R	CM	M	
3.3.4	버스 기능 모델	VHDL,Verilog	R	R	R	
3.3.5	Bonded out VC/Prototype		R	R	R	
3.4	<b>논리 설계정보</b>					
3.4.1	합성가능한 RTL소스	합성가능한 구조 (VHDL,Verilog)	M	CM	CR	Netlist could be used
3.4.2	합성 제약조건	Synopsys DC Shell	M	CM	-	

절	Deliverable	현재 사용되는 형식	Soft VC	Soft VC	Soft VC	비고	
3.4.3	Floor planning	Floorplanning shell	ASCII, LEF	M	M	M	Inter-block
		Floorplanning 제약조건	PDEF,LEF,SDF	-	CM	-	Inter-block
3.4.4	Netlist	VHDL,Verilog, EDIF	-	CM	CR	Synthesizable RTL could be used	
3.4.5	기본 지연 모델	SPIICE	R	M	M	No current state dependent block format	
3.4.6	외부 연결 모델	TLF,Synopsys NLDM, ITL,MMF	-	CR	CM		
3.5	<b>테스트 정보</b>						
3.5.1	테스트 인터페이스 구조		M	M	M		
3.5.2	테스트 방법	Document	M	M	M		
	Open-box 테스트 명시	Document	M	CM	-	If the VC faults fail into open box category	
	Black-box 테스트 명시	Document	-	CM	M		
	테스트 패턴	WGL,VCD,ASCII	-	CM	M	Dependent upon Test Method	
	BIST	WGL,VCD	CM	CM	CM	Large memory arrays	
	지연 테스트	WGL,VCD	-	R	R		
	신뢰성	WGL,VCD	-	R	R		
	오류분석 및 격리	WGL,VCD	-	R	R		
3.6	<b>레이아웃설계 정보</b>						
3.6.1	블록 사양	GDSII,LEF,DEF, SPIICE	-	CM	CM	Estimated for Soft and Firm	
3.6.2	핀 정보	LEF,DEF	-	CM	M	Required if hard is netlist based	
3.6.3	공개 및 방어정보	LEF,DEF	-	CM	M		
3.6.4	Footprint	LEF	-	CM	CM		
3.6.5	전원/그라운드	LEF,DEF,ASCII	-	CR	R		
3.6.6	전원모델	TBD	R	R	R		
3.6.7	물리적 넷리스트	SPIICE	-	CM	M		

VSIA에서 규정하고 있는 IP Deliverables의 내용을 보면 크게 IP의 일반적인 기술정보를 나타내는 사용자 지침(User Guide)과 시스템 내에서의 IP의 활용을 나타내는 시스템 구조(System Architecture) 및 시스템 설계(System Design) 부분 그리고 IP의 기술적인 실제 활용방법을 나타내는 논리 설계(Logic Design), 테스트 정보(Test Requirement) 및 레이아웃 설계 정보(Physical Block Implementation)로 크게 구분할 수 있다. 이 중 논리설계, 테스트 정보 및 레이아웃 설계 정보 부분은 해당 IP가 VSIA에서 구분한 3가지 IP(soft, firm, hard) 중 어느 부분에 해당하는가에 따라 실질적인 전달물 규정이 크게 차이가 난다.

## 1. 사용자 지침 (User Guide)

IP에 대한 사용자 지침(User Guide)은 크게 IP 사양(Specification), IP 정보(Claims & Assumption), IP 정보에 대한 검증(Verifications of Claims), IP의 버전 정보 및 오류정보(Version History & Known Bugs) 그리고 응용노트(Application Notes)로 구분된다. 이러한 사용자 정보는 IP를 활용하는 데 있어 가장 기본이 되는 정보로 모든 IP 개발자들은 IP 사용자들에게 반드시 제공해야 한다. 일반적으로 어떤 유/무형의 제품을 제품 사용자에게 전달할 때 반드시 필요한 사용자 설명서와 같이 IP에 대한 사용자 지침은 그 IP를 활용하는데 있어 사용자가 별도의 문의사항 없이도 해당 IP를 충분히 활용할 수 있도록 상세하게 작성되어야 하며, 그 IP를 활용하는 데 있어 부가적으로 요구되는 각종 파일 및 문서들이 어떠한 형태로 구성되어 있으며 어떠한 형식으로 활용이 되는 것인지 등에 대해 충분히 작성되어야 한다. 실제로 일반 IP 사용자들은 국내외를 막론하고 해당 IP에 대해 가장 신뢰를 확인할 수 있는 부분이 바로 해당 IP의 사용자 지침을 보고난 후라는 것을 볼 때, IP 사용자 지침은 IP를 나타내는 얼굴이라 볼 수 있다.

이러한 사용자 지침의 내용은 IP를 개발한 후 활용에 이르기까지의 IP 버전 정보를 비롯하여 IP의 버전관리가 어떻게 진행되어 왔으며, 각 버전간의 차이점이 무엇인지를 명확하게 표현해 주어야 한다. 특히 응용노트(Application Notes)에는 어떠한 방법으로 활용할 수 있는지에 대한 정보를 명확히 하여 IP 사용자들이 IP 개발자에게 추가적인 요청사항이 발생하지 않도록 작성하여야 만이 IP 개발자들의 기술지원 내역을 줄이고 사용자들의 부담을 덜어줄 수 있는 좋은 IP가 될 수 있다.

사용자 지침의 내용 중 사양(Specification)에 대한 정보는 해당 IP를 선택한 사용자들이 가장 관심을 가지는 부분으로 그 IP의 기술적인 사양 뿐 아니라 내부구성에 대한 정보까지도 담고 있어야 한다. 이러한 기술사양은 해당 IP를 선택하는 기준이 되기도 하는데 일반 IP 사용자들이 자신들이 원하는 사양에 맞는 IP를 고르는 기준으로 활용되기 때문이다. 이러한 사양(Specification)에는 IP의 시스템에서의 활용정보를 비롯하여 블록도, 클럭정보, I/O와 관련된 타이밍 및 인터페이스 정보 등이 반드시 포함되어야 한다.

## 2. 시스템 구조(System Architecture)

시스템 구조에서는 해당 IP를 시스템 내에서 어떻게 활용할 수 있는지를 나타내는 정보로 시스템 검증 모델(System Evaluation Model)에 대한 내용을 포함하여야 한다. 이러한 모델은 해당 IP를 활용한 SoC가 시스템 내에서 어떠한 함수적 능력을 가지며 어떠한 성능을 발휘할 수 있는지 등에 대

해 기술되어야 한다. 어떤 의미에서 보면 이미 IP 개발자가 일부 시스템을 만들어 IP를 검증하면서 그 해당 시스템을 기준으로 한 성능정보를 기술한다는 측면과 유사(이러한 시스템 검증 정보를 가지고 있을 경우 응용노트를 작성하기 수월하다)하다고 볼 수 있으며, 이러한 정보는 단일 IP의 활용적인 측면보다 해당 IP를 SoC화 했을 때 어떻게 활용이 가능한지를 나타내게 된다.

시스템 구조는 실질적으로 볼 때 IP 자체에 국한된 모델이 아니며, 따라서 VSI의 규정 내에서도 매우 간략하게 표현되어 있는 것이 사실이다. 그러나 향후 SoC의 개념이 결국 시스템 기술을 포함하는 측면으로 확대된다고 볼 때 현재 IP의 범위를 초과하는 시스템 수준의 IP 정보를 포함하는 것이 향후 IP를 활용한 SoC 개발을 합리적으로 연결할 수 있는 타당한 방법일 것이다.

## 3. 시스템 설계 정보(System Design)

시스템 설계에 대한 정보는 '3.2 시스템 구조'와 같이 현재의 IP 기술에서는 크게 중요시 되고 있지 않은 부분이다. 이 정보는 IP 자체의 정보라기 보다는 역시 IP를 활용한 시스템의 정보를 나타내는 것이 주 목적이며, IP를 활용한 시스템 내에서의 프로세서 구조, 버스 구조 등에 대한 정보를 나타낸다. 이러한 정보는 SoC 개발자가 필요한 정보이기 보다는 SoC를 활용한 시스템 기술자들에게 더욱 필요한 정보이며, 결국 IP를 활용한 SoC를 사용하는 최종 사용자가 시스템 개발자라는 측면에서 볼 때, 시스템 수준의 개발 환경이 확실하게 지원되는 시점에서 부각될 것으로 보인다.(최근 이미 시스템 수준의 설계환경이 점차 보급되고 있으며, 이를 활용한 성공사례들이 나타나고 있어 IP를 활용하기 위한 규정보다는 시스템을 위해 IP가 가져야 할 규정으로 확대되는 것이 타당할 것이다)

## 4. 논리 설계 정보(Logic Design)

논리설계 정보의 주된 핵심은 Soft IP를 위한 정보이다. 그러나 Soft IP로부터 Firm IP가 생성된다고 볼 때, 논리설계정보는 2가지 구분의 IP에 적용되는 주요 핵심사항이라고 할 수 있다. 논리 설계 정보는 주로 HDL, Verilog 기반의 설계정보를 비롯하여 이러한 설계정보로부터 어떻게 RTL 수준의 데이터까지 최종적으로 구축하는가에 대한 정보이다. 따라서 논리설계 정보는 원시코드(source code)를 비롯하여 합성에 필요한 정보(synthesis script 등), Floorplanning 정보, Firm IP의 중요 요소인 Netlist 정보, 지연 정보 등을 모두 포함하고 있다.



그러나 최근 Soft IP와 Firm IP 간의 구분이 점점 흐려지고 있으며 Soft IP 이전에 C/C++ 등의 일반적인 프로그래밍 언어를 활용한 Model IP 등을 활용하여 직접 SoC 개발을 수행할 수 있는 여건이 구축되면서 Firm IP를 Soft IP에 포함시키는 분위기도 나타나고 있다.

논리설계 정보 중 Netlist, 지연모델(Delay Model)과 외부 연결모델(Peripheral Interconnect Model)은 주로 Firm IP와 Hard IP를 위한 정보로 활용된다.

지연모델의 경우 Firm IP에서는 상대적 위치정보와 연결 관계가 Firm IP 내에 포함되어 있기 때문에 시스템 레벨 타이밍 분석을 위한 필요한 정보가 충분히 포함되어 있어야 하며, Hard IP인 경우에는 트랜지스터 레벨 정적 타이밍(Static Timing) 분석 정보(온도, 공정, 전압변화에 대한 특성데이터 추출 및 curve fitting)가 포함 되어야 한다.

외부 연결 모델은 내부 블록간의 인터페이스를 표현한 것으로 블록 고유의 지연을 주변에 연결된 RC(저항과 캐패시턴스)와 분리하기 위해 필요하다. 이렇게 I/O로부터 게이트까지 연결관계를 유지함으로써 상위 계층에서 지연 시간을 계산할 경우 부정확하게 근사되어 있는 부하나 연결관계를 활용할 때 보다 실제의 연결관계를 사용하여 계산할 수 있다. 외부연결모델은 I/O 연결관계가 명확한 Hard IP에는 반드시 포함되어야 하는 사항이다.

## 5. 테스트 정보(Test Requirement)

테스트 정보는 IP를 활용하는 데 있어 최종 동작의 검증을 어떻게 수행할 것인가를 결정짓는 중요한 사항이다. 이 정보는 반드시 모든 구분의 IP에 대해 제공되어야 하며, 매우 명확하고 정확하게 기술되어야 할 뿐 아니라 IP 사용자가 이 정보를 기본으로 사용자가 희망하는 테스트를 수행할 수 있을 만큼 충분한 정보가 전달되어야 한다. 또한 SoC 설계를 위해 메모리가 활용될 경우 이 메모리 셀을 어떻게 테스트 할 수 있는가에 대한 방법도 반드시 포함되어야 한다.

테스트 정보에는 기본적인 테스트 방법을 비롯하여 Open & Black-box 테스트, 테스트 패턴, 메모리 셀을 위한 BIST 등의 정보를 포함하여야 하며, Firm IP나 Hard IP를 위한 신뢰성(Reliability) 및 오류 분석/격리(Failure Analysis & Isolation)에 관한 정보도 포함하여야 한다.

## 6. 레이아웃 설계 정보(Physical Block Implementation)

레이아웃 설계 정보는 Firm IP와 Hard IP를 위한 정보이다. Soft IP의 경우에는 별도의 물리적 블록(Physical Block)을 사용하지 않기 때문에 IP를 SoC화 하는 관점에서 볼 때 최종단계에 이르기 위한 정보라고 할 수 있다.

Hard IP에 있어서 레이아웃 설계정보는 높은 수준의 물리적 구조를 가지는 IP와 다른 IP들 간의 통합에 필요한 블록 특성을 기술한 내용으로 레이아웃 설계정보에는 블록 사양, 핀 정보, 공개 및 방어(Porosity & Blockage) 정보, Footprint, 전원 및 물리적 Netlist 정보 등이 포함된다.

블록 사양은 GDSII 형식으로 된 IP의 레이아웃 관련 상세정보이며, 이 정보는 시스템 레벨에서 설계규칙의 검증을 위해서 사용된다. 핀 정보에는 크기, 치수, 포트 등에 관한 정보, 배치, 배선, 기생 파라미터 추출 등을 위한 grid 구조에 관한 정보가 포함되어 있으며, SoC 개발을 하면서 메모리 셀을 점차 많이 사용하고 논리 소자의 수가 증가함에 따라 crosstalk, 기타 노이즈 관련 문제점 등이 발생하여 배치배선이 불가능한 영역과 자유롭게 배선 가능한 영역을 기술하기 위한 공개 및 방어정보가 함께 포함된다.

전원/그라운드(Power & Ground) 정보는 구현하고자 하는 SoC의 경계에 대한 전원과 그라운드 연결정보가 포함되어 있어야 하고 칩 레벨에서 적정 수준의 전압강하 및 신뢰도 보장을 위하여 전류 밀도 정보가 포함되어야 한다. 그리고 전압강하 노이즈, 최악조건 하에서의 동시 switching 관련 제약조건 등이 명시되어야 만이 최종 SoC 개발을 위한 실질적인 정보로 활용될 수 있다.

## 기타 VSIA의 관련 권고사항

앞에서 논의한 VSIA의 일반적인 IP 전달물 규정과는 별도로 VSIA에서는 몇 가지 권고사항(Guidelines)에 대해 함께 제공하고 있다. 이 권고사항은 IP를 사용하는데 있어 필요한 전달물과 같은 내용이 아니라 개발한 IP에 대해 수정/보완이 용이하고 IP 사용자가 IP 블록이 내포하고 있는 해당 의미를 정확하게 파악할 수 있도록 돕기 위한 방편으로 제공된다. 이러한 권고사항에는 IP를 나타내는 이름을 비롯하여 원시코드의 구분, I/O 구분 등을 용이하게 하기 위한 IP 이름에 대한 사항(Naming Guideline)을 비롯하여 테스트 권고사항(Test Guideline), SoC와 IP의 계층적 구조결합을 위한 권고사항(Chip to IP Hierarchical Integration) 등이 있다. 이러한 권고사항에 대해서는 VSIA의 홈페이지(<http://www.vsi.org>)에 접속하여 Architecture Document를 다운로드 받아 확인할 수 있다.

## Model IP

앞서 언급한 바와 같이 VSIA의 IP 구분이나 규정에는 Model IP에 대한 내용은 존재하지 않는다.

어떻게 보면 VSIA IP 구분 중 Soft IP 내에 Model IP가 포함될 수 있으나 현재까지의 내용이나 IP 구분으로 볼 때 아직까지 알고리즘 수준의 데이터나 EDA 환경이 아닌 개발환경의 데이터를 나타내는 Model IP에 대한 내용은 VSIA의 규정이 포함시키는 것은 불가능하다. Model IP에 대해서는 IPCoS에서 IP 표준안을 개발하면서 SoC 개발을 위한 IP의 범위를 시스템 수준까지 포함하면서 정의하기 시작한 것이 처음이라고 할 수 있을 것이다.

이러한 Model IP는 모든 반도체 설계 이전에 시작하는 C/C++ 등을 활용한 알고리즘 검증이나 MathLab 등과 같은 알고리즘 시뮬레이터 등을 포함하는 각각의 환경이 가지고 있는 모든 언어를 포함하고 있다. 이러한 각각의 개별적인 환경의 언어들은 EDA 환경에서 바로 사용이 가능하도록 지원되지 않으나 과거 EDA 환경에서 직접 활용이 불가능하였던 C/C++ 언어의 알고리즘 설계 데이터가 최근에는 HDL 수준의 언어로 직접 변환이 가능하거나 또는 Physical 합성의 단계까지 지원되는 현실을 감안할 때 향후에는 이러한 환경의 지원을 받을 수 있으리라고 생각된다. 또한 위에서 언급한 VSIA 전달물 규정의 시스템 구조(System Architecture, 3.2 항목) 및 시스템 설계정보(System Design, 3.3 항목)에서 볼 수 있듯이 현재에는 VSIA에서 큰 비중을 차지하지 않고 있는 시스템 수준의 설계에 대한 관심이 향후 중요한 변수로 작용할 수 있기 때문에 이 부분에 대한 정의가 필요할 것이다.

아래의 표는 현재 IPCoS에서 정의하고 있는 Model IP와 관련된 전달물 규정으로 각각의 Model IP를 구분하고 Model IP에 적용되는 각각의 전달물들에 대한 내용을 규정한 것이다.

Model IP의 구분	전달물 규정 항목번호					
	5.1	5.2	5.3	5.4	5.5	5.6
C/C++ 수준의 Model	M	CM	M	CM	M	CM
개별 제품의 언어 Model	M	CM	M	R	CM	R
Embedded 프로그램 Model(OS 등)	M	M	CM	-	-	-

## 1. 사용자 지침

각각의 모든 Model IP는 반드시 사용자 지침을 포함하여야 한다. 이는 IP의 개념에서 IP 사용자가 그 해당 IP를 활용하는데 있어 기본적인 정보를 획득할 수 있도록 하기 위한 기본적인 규정이다.

## 2. 시스템 구조정보

Model IP는 그 구분에 따라 시스템 수준의 검증이 가능한 모델이일 경우

반드시 시스템 구조 정보에 대한 전달물을 제출하여야 한다. Embedded 프로그램 Model의 경우에는 하드웨어 기반의 프로그램 모델이기 때문에 반드시 시스템 구조 정보를 작성하여야 한다.

## 3. 시스템 설계정보

시스템 설계정보 중 검증용 테스트 벤치는 모든 Model IP들이 제공하여야 하며, Model IP의 특성을 감안할 때 하드웨어 테스트가 가능하도록 I/O에 대한 프로그램을 별도로 제공하여야 한다. 또한 Model IP를 활용할 수 있는 구현 시스템의 설계 정보를 제공하여야 한다.

## 4. 논리설계 정보

C/C++ 기반의 Model IP와 개별 제품의 언어로 표현된 Model IP는 향후 SoC로 활용될 수 있는 경우(EDA 환경의 지원이 허락하는 한도 내에서) 반드시 합성 가능한 RTL 소스를 제공하여야 하며, 이에 따른 합성계약조건도 함께 지원되어야 한다.(C/C++ 환경의 Model IP는 가능한 한 합성가능 원시코드를 지원하는 것을 기본으로 한다) 특히 개별 제품의 언어로 기술된 Model IP의 경우 C/C++ 언어로의 변환이 가능할 경우 EDA 환경의 지원을 받아 SoC로 활용이 가능하기 때문에 가능한 범위 내에서 모듈화를 통해 합성이 가능하도록 기술하는 것을 원칙으로 한다.

## 5. 테스트 정보

Embedded 프로그램 Model은 5.2 및 5.3 항목을 통해 5.5 테스트 정보를 대신한다. C/C++ 수준의 Model은 컴퓨터상에서 하드웨어 I/O를 활용하여 하드웨어 테스트가 가능하도록 외부 인터페이스를 구현하여 테스트하여야 하며, 이를 반드시 테스트 인터페이스 구조에 명시하여야 한다. 개별 제품의 언어 Model의 경우 소프트웨어적인 데이터를 하드웨어를 사용하여 입출력 시킬 수 있을 경우(실시간이 아니어도 무방함)에는 그 외부에 하드웨어를 연결하여 입출력을 테스트할 수 있는 정보를 제공하여야 한다.

## 6. 레이아웃 설계 정보

6. 레이아웃 정보는 4. 논리설계 정보와 마찬가지로 EDA 환경이 지원하는 범위 내에서 5.4의 규정을 따른다.



## 기타 VSIA의 IP 전달물 규정

VSIA에서는 기본적인 전달물 규정을 정의한 Architecture Document 외에 각 Working Group별로 각각의 세부 규정에 대한 상세한 Deliverable을 규정하고 있으며, 이러한 세부 규정들은 현재 VSIA의 회원에 대해서만 제공/지원되고 있다. VSIA의 회원이 아닌 경우에는 별도의 비용을 부담하고 VSIA로부터 규정을 구입할 수 있으며, 회원으로는 기관 뿐 아니라 개인도 가입이 가능하다. 자세한 VSIA의 IP와 관련된 전달물 규정에 대해서는 VSIA의 홈페이지를 참고하기 바란다.

## 결론

IP의 전달물 규정에 대해 세계적으로 공통되어 적용되는 규정은 아직 존재하지 않으며 국내에서는 몇 개 IP 관련 기관이 나름대로의 규정을 작성하여 배포한 사례가 있다. 그러나 현재 VSIA에서 가장 활발하게 움직이며 IP와 관련된 규정을 제정하고 있으며, 이 Alliance에 많은 반도체 관련기관들이 함께 참여하고 있어 우선적으로는 VSIA의 전달물 규정을 준수하는 것이 향후에도 많은 도움이 될 것으로 생각된다. 특히 최근에는 Cadence Design Systems, Synopsys, Mentor Graphics, ARM, ST-Micro, Philips 등이 SPIRIT(Structure for Packaging, Integrating and Re-use IP within Tool-flows)라고 하는 연합체를 구성하여 EDA 환경 내에서의 환경 지원표 준화를 위해 협력하고 있어 향후 귀추가 주목된다.