

ASIC 설계 기술

金春慶, 徐仁煥

金星일렉트론(株) 半導體研究所

최근 우리나라의 세계 반도체 시장 점유율이 갈수록 높아지고 있다. DRAM과 같은 경우는 세계 2위 수출국으로 떠올랐다. 이에 따른 선진 외국의 무역 압력이 거세지고 있다. 특히 미국의 경우 미·일 반도체 협정 체결을 한국에까지 확산하려고 하며, 각종 기술에 대한 특허 제소를 하고 있다. 그러나 국내 반도체 산업은 DRAM 부분에 집중되어 있어서 기술 집약적인 ASIC 부분은 취약하다. 또한 세계의 반도체 시장중에서 ASIC이 차지하고 있는 비중이 갈수록 커지는 현상에 미루어 보아 빠른 시일내에 ASIC 부분을 강화해야 할 필요성이 있다.

본고에서는 국내에 도입된 ASIC 설계 환경을 살펴본 후, 현재 추진되고 있는 새로운 ASIC 설계 환경을 고찰한다. 또한 ASIC 설계 환경에서 중요한 역할을 하는 ASIC CAD 환경에 대하여 살펴보기로 한다.

I. 서 론

반도체 제조기술이 발전함에 따라 TTL 소자들의 조합으로 구성된 시스템을 하나의 집적회로 chip으로 구현하고자 하는 요구가 발생하였다. TTL 시스템을 하나의 집적회로 chip으로 구현함으로써 시스템의 기능을 외부로부터 보호할 수 있게 되었으며, 시스템의 속도 향상에도 기여하였다. 뿐만 아니라 하나의 IC가 다수의 TTL을 대신함으로써 PCB의 크기를 줄이는 등의 비용 절감 효과를 가져왔다. 또한 집적 기술이 발전함에 따라 수백만 게이트를 하나의 IC에 집적하게 되었다.

최근의 동향은 고속의 연산 기능을 가지는 복잡한 회로를 빠른 시간에 제품화하는 것이다. 즉, 제품의 life cycle이 짧아졌다. 이와 같은 현상은 가전제품, 정보기

기 제품 및 산업용 제품 전반에 걸쳐서 나타나고 있다. 특히 컴퓨터에 관련된 제품의 life cycle은 눈에 띄게 짧아졌다.

초기에 ASIC 설계 기술을 보면, 사용자가 원하는 기능을 TTL 소자를 이용하여 bread-board에서 구현하여 동작을 확인하고, 이에대한 layout을 mylar sheet에 제도하였다. 그런 후에 이를 mask로 제작하여 반도체를 제조하였다. 그러나 컴퓨터의 발전과 이에따른 소프트웨어의 개발에 힘입어 layout editor, logic simulator 등의 CAD tool을 이용하게 되었다. 최근에는 ASIC 설계자가 보다 시스템 개발에만 치중할 수 있도록 한 CAD tool이 개발되었다.

II. Layout 설계 방법에 따른 ASIC 분류

ASIC은 layout 설계 방법에 따라 표 1 에서와 같이 크게 2 가지로 (non cell based IC, cell based IC) 분류된다. Non cell based 방식은 수작업에 의한 layout 설계 방식으로 F/C (full custom) 방식이라 한다. 이 방식은 chip의 크기는 설계자의 능력에 따라 매우 작게까지 구현할 수 있다. 그러나 수작업에 의한 설계이므로 설계 기간이 매우 길고, 수작업에 따른 설계 오류의 발생 가능성이 매우 크다. 또한, 설계 기간이 길어 적기 시장 출하가 어려우므로 좁은 market window를 가진 제품 설계에서는 경쟁력이 없는 설계 방식이다. Cell based IC 설계 방식은 다시 S/C (standard cell) 방식과 G/A (gate array) 방식으로 분류된다. S/C와 G/A 모두 특정한 논리를 구현하는 cell을 미리 layout하여 cell library에 준비한다. Cell library의 모든 cell은 일정한 높이로 설계되어 있어 상호 인접하는 것 만으로도 전력

선 (VDD, VSS)을 공유할 수 있다. 준비된 cell들은 논리 회로도에서 구현된 cell과 1:1 대응 관계를 가진다. 그러므로 준비된 논리 회로도에 따라서 cell들을 적당히 배치를 한다. Cell을 배치 할 때에는 열 또는 행을 맞추어 배치한다. 이를 cell row라 한다. 또한 cell row와 cell row 사이에 배선을 할 수 있는 공간을 확보한다. 이를 channel이라 한다. 배치가 끝나면 논리 회로도의 연결정보에 의해서 cell의 각 pin을 연결한다. S/C과 G/A의 큰 차이점은 다음과 같다. S/C은 cell을 배치할 수 있는 곳이 정하여져 있지 않다. 따라서 channel의 크기가 일정하지 않다. 반면에 G/A는 cell이 배치될 수 있는 곳이 지정되어 있다. 즉, 모든 channel의 크기가 일정하다는 것이다. 따라서 S/C은 100% 배선을 할 수 있지만, G/A는 channel 영역이 모자라는 특정 부분(주로 중앙 부분)에서 미결선이 생길 수 있다. G/A는 channelled G/A 와 SOG (sea of gate)로 세분된다. Channelled G/A는 cell row와 channel이 교대로 있지만, SOG는 cell row만 있다. 그러므로 SOG에서의 배선은 cell 내부에 비어있는 공간에서 배선을 하게 된다. 일반적으로 SOG는 집적도를 높일수 있으나 배선 기법이 복잡할 뿐 아니라 배선에 사용되는 layer가 2 이상일 때 더 효과적이다. 또한 G/A에는 FPGA (field programmable G/A)가 있다. 일반적인 FPGA의 구조는 그림 1에서와 같이 3가지의 기본 cell로 구성되어 있다. FPGA를 둘러싸고 있는 프로그램 가능한 I/O 블럭과 구성가능한 (configurable) 논리 기능 단위가 내부에서 배열 구조를 이루고 있다. 사용자는 논리 기능을 이들 I/O 및 논리 기능 단위를 이용하여 구현한다. 그러므로 I/O 및 논리 기능 단위의 수효는 FPGA의 용량을 좌우한다. FPGA의 또다른 기본 cell은 프로그램 가능한 연결 소자이다. 이 소자는 I/O 및 기본 논리 기능 cell 사이의 channel을 채운다. 또한 이 연결 소자는 임의의 연결을 할 수 있다. 현재까지 상용화된 FPGA의 연결 방식은 static memory 기술을 사용하는 것과 anti-fuse 기술을 사용하는 것이 있다. 전자의 경우는 설계의 변경이 있는 경우에 추가 비용없이 수정이 가능한 장점이 있다.

I/O cell이 IC 가장자리에 위치하여 있고, logic block의 배열이 중앙에 있다. 사용자의 논리 기능은 이들 I/O 및 logic block으로부터 만들어진다. 그러므로 I/O 및 logic block의 수효는 IC의 논리 기능의 크기를 좌우한다.

오늘날의 cell based IC 설계 방식에는 macro function을 수행하는 임의의 크기와 모양을 가진 cell을

library에 등록하여 사용하기도 한다. 이로 인하여 하나의 block에는 macro block과 cell이 혼재하게 되었다. 또한 F/E 설계에서 사용된 계층구조를 이용하여 B/E 설계에서 bottom-up 방식의 설계를 한다.

표 1. ASIC의 분류

분 류	설계기간	설계비용	기 타
Non-Cell Based	F/C (6개월이상)	길다 대규모 대량생산에 유리	수작업에 의한 설계 오류발생 가능성이 크다
Cell Based IC	S/C	비교적 짧다 (10-12주)	중규모 생산 에 유리
	G/A	짧다 (6-8주)	소규모 생산 에 유리
	FPGA	매우 짧다 (1일 이내)	Sample 생산 에 유리
			자동 배치 배선 tool을 사용하여 설계 자동 화를 한다.

Layout 설계에서 생산까지의 기간 및 설계 비용 면에서 보면 F/C 방식은 오랜 설계 기간이 필요한 반면 대량 생산시에는 최소의 비용이 든다. S/C 방식의 설계는 비교적 짧은 기간에 설계가 가능하다. 또한 대량 생산시에 잇점이 있다. G/A의 경우는 짧은 기간에 설계가 가능하며 비용도 저렴하다. 왜냐하면 반제품에서 몇가지의 공정만 추가되기 때문이다. FPGA의 경우는 논리 회로 설계가 끝난 직후에 제품 출하가 가능하지만 단위 비용이 많이 든다. 그러나 견본 출하, 또는 초기의 시장 점유율 확대등의 목적에는 유리하다. 특히 FPGA는 CMOS

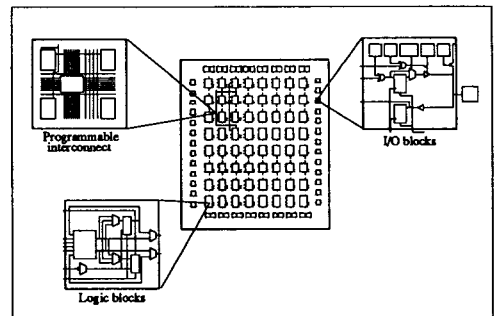


그림 1. FPGA의 구조

SRAM 기술과 함께 발전하므로 chip당 집적도가 증가되고 있는 추세에 있다. 이는 수년안에 FPGA가 현재의 channelled G/A를 대응 할 수도 있다는 것을 의미한다.

III. ASIC 설계자가 설계이전에 고려해야 할 사항

80년대 후반기 부터 반도체를 이용한 전기, 전자제품의 수명이 점차 짧아지고 있다. 이에따라 제품에 소요되는 IC의 수명 또한 단축되고 있는 추세이다. 이와같은 요인으로 인하여 “신제품의 납기가 6개월 지연되면 그 제품의 수명기간 동안의 순이익은 1/3 감소한다”는 연구 보고가 있다. 그러므로 설계자는 최적의 시기에 최적의 비용으로 제품을 출하할 수 있도록 노력해야 한다. 이를 위해서 설계자는 설계이전에 다음 3가지 사항을 고려해야 한다.

- B/E 설계 방식을 무엇으로 할 것인가?
- 어떠한 CAD tool을 어떻게 사용할 것인가?
- Package 는 가능한가?
- One-pass success를 하기 위한 방안은 무엇인가?

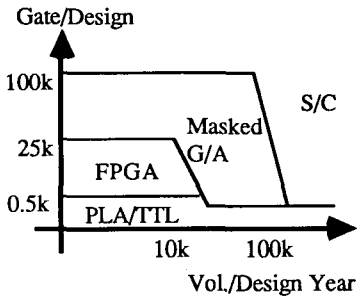


그림 2. Gate 집적도와 소요 물량에 따른 설계 방법

B/E 설계 방식의 선택은 제품의 출하 시기와 제품의 단위 비용 결정에 중요한 영향을 끼친다. 그림 2에서 연간 소요되는 물량과 gate 집적도에 따른 유효한 설계 방식을 보였다.

그림 3에서는 출하 시기와 gate 집적도에 따른 설계 방식의 선택에 관하여 보였다. 그림2 와 그림 3에서 설계자는 gate 집적도와 출하시기 및 비용을 고려하여 설계 방식을 선택하여야 함을 보여 준다.

또한 설계에 있어서 어떤 CAD tool을 사용할 것인가를 잘 결정하여야 한다. 같은 기능을 가진 CAD tool이라 할지라도 각각의 tool에 따른 장·단점이 있다. 예를 들어 P&R (placement & routing) tool을 선택하고자 하는 경우에 어떤 tool은 빠른 시간에 작업을 완료하지만 timing driven layout 을 제공하지 못하는 반면, 다른 tool은 작업시간은 많이 소요되나 timing driven layout 이 제공된다고 하자. 그리고 설계된 회로는 timing에 매우 민감하다면 설계자는 후자의 tool을 선택하여야 한다. 즉, 설계자는 CAD tool의 특성과 회로의 특성을 잘 파악하여 각 설계 단계에서 적합한 tool을 사용하도록 미리 결정하여야 한다. 이때 tool 사이의 interface 가능성을 반드시 함께 조사하여야 한다. 그리고 I/O pin의 수와 IC의 크기를 예측하여 package에 어려움이 없도록 하여야 한다.

앞서 설명된 모든 것이 고려된 설계였다고 하여도 one-pass success에 실패한다면 납기를 지연시키게 된다. 그러므로 delay, fanout, noise, heat 등의 설계과정에서 발생하는 문제점 뿐 아니라 실장에 있어서 발생할 수 있는 문제점을 파악하여 대처해야 한다. 이와 같은 사항은 주로 경험에 의한 know-how 축적이 중요하다. 또한 설계자는 제조후에 IC를 test 하기 용이하도록 설계하여야 한다. Testable 설계는 최악의 경우, 설계에서 실패하였을 때에도 고장진단에 요긴하게 사용된다.

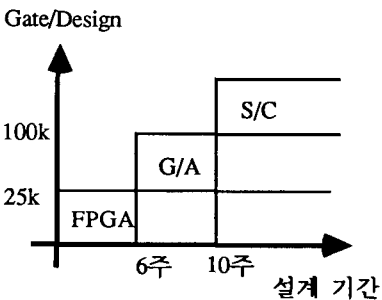


그림 3. Gate 집적도와 설계 기간에 따른 설계 방법

IV. 현재 보편적으로 사용되고 있는 ASIC 설계 과정

현재 ASIC 설계는 그림 4에서와 같이 top-down 방식을 많이 사용하고 있다. Top-down 설계는 먼저 시스템의 사양을 결정하고, 이를 만족하는 logic diagram 을 schematic capture를 통하여 구현한다. 구현된 논리회로도는 logic simulator를 이용하여 회로도의 기능을 검증한다. 검증된 회로도는 다양한 layout 방식을 이용하여 설계된다. 생성된 layout 은 제조 기술과 관련된 설

계 규칙 검사등을 한다. 그리고 layout으로 부터 delay time을 추출하여 최초의 논리회로 검증에서 사용된 logic simulator에 feedback하여 시간지연에 따른 오동작 여부를 검증한다. 이를 back-annotation이라 한다. 검증이 확인되면 제조하여 test 한 후 출하한다.

ASIC 설계에서 layout 설계 이전까지의 단계를 F/E (front end) 설계라 하고 layout 부터 test 까지를 B/E (back end) 설계라 한다.

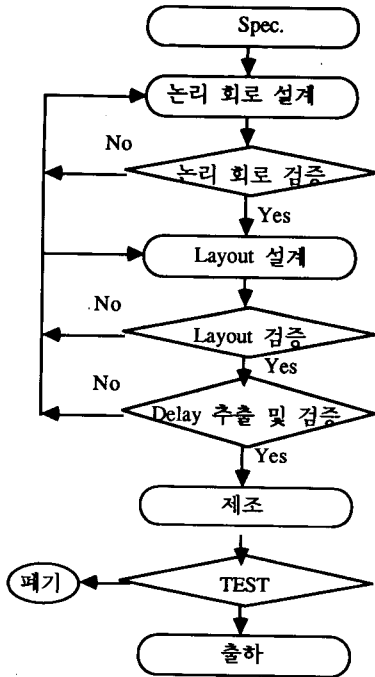


그림 4. ASIC 설계흐름

초기 국내의 ASIC 설계는 주로 국외에서 생산된 IC를 복제하는 수준이었다. 그러나 IC 당 집적도가 증가되고, 시장의 변화가 빨라지면서 자체 설계를 하지 않으면 안되었다. 자체 설계에서의 위험 부담은 논리회로의 구현 가능성 여부 보다도 critical path delay, setup/hold time delay, back-annotation 등과 같은 timing과 test에 관한 문제였다.

1. Front End 설계

F/E 설계는 크게 2부분으로 구별된다. 하나는 논리회로 설계과정이고 다른 하나는 논리회로 검증 과정이다. 논리회로 설계는 schematic capture tool을 이용하여 사용자가 원하는 기능을 구현한다. 설계자는 설계

사양을 검토하고 이를 구현할 B/E 설계 방법을 결정한다. B/E 설계 방법의 결정은 schematic diagram에서 사용되는 cell library가 결정됨을 의미한다. 논리 설계가 끝나면 이를 검증한다. 초기에 사용된 검증 tool은 logic simulator가 대부분이었다. 최근에는 sub micron 공정의 개발과 고속 소자의 등장에 따라 timing에 관련된 검증이 중요하게 되었다. 따라서 논리 검증시에도 setup/hold time, clock skew 등이 반드시 고려되어야 한다. 또한 critical path delay는 전체 시스템의 성능을 가능하는 중요한 요소이므로 timing simulator를 통하여 검증한다. 이 외에도 설계 회로의 크기가 증가함에 따라서 test가 용이하지 않게 되었다. 특히, 입출력의 수가 증가함에 따라 모든 경우의 수에 대하여 test하는 것은 불가능하게 되었다. 따라서 설계 초기 단계에서 test를 고려하도록 한다. Test에 관하여서는 5장에서 설명하였다.

2. Back End 설계

Layout 설계는 앞서 설명한 것과 같이 F/C, S/C, G/A 방식이 있다. S/C 및 G/A 방식에서 사용되는 CAD tool로는 automatic P&R (placement & router)이 있다. 이 tool의 목적은 F/E 설계에서 사용된 cell을 자동 배치 및 배선을 하는 것이다. 또 다른 B/E tool로 layout editor가 있다. 이는 layout을 직접 설계하거나 설계된 layout을 수정하는데 사용된다. 또한 symbolic layout editor를 이용하면 object 방식으로 layout을 설계 및 수정할 수 있다. P&R tool에서 사용될 cell은 module generator를 이용하여 설계되기도 한다. Module generator는 공정기술의 변화에 능동적으로 대처할 수 있어서 많은 관심을 모으고 있다. Layout 설계가 끝나면 layout 검사를 한다. 여기에는 DRC (design rule check), ERC (electric rule check) 및 LVS (layout vs. schematic) 과정이 있다. 최근에는 보다 정밀한 back-annotation을 위하여 LPE (layout parameter extract) 기능이 첨가되었다.

V. TEST

회로의 집적도가 증가함에 따라서 입출력의 pin 수가 증가하였다. Pin 수의 증가는 이를 test하기 위한 pattern 수의 증가를 의미한다. Test pattern의 증가 속도는 pin 수에 대하여 지수 함수적으로 증가하므로 test 불능의 상태에 이르게 되었다. 이와 같은 사항을 해결

하기 위하여 IBM에서는 test가 용이한 LSSD (level sensitive scan design) 방식을 채택하여 모든 설계에 사용하고 있다. 이처럼 이제 test는 설계에서 중요한 요소가 되어 설계에서 반드시 고려되어야 할 사항이 되었다. DFT (design for testability) 용어의 등장이 이를 입증해 준다.

일반적으로 test에는 exhaustive test, structured test 및 functional test로 분류된다. Exhaustive test는 모든 입력에 대한 조합 가능한 test pattern에 대하여 test하는 가장 단순한 방법이다. 입력의 숫자가 작을때에 사용하던 방법이나 오늘날과 같은 대규모 회로에는 적합하지 않다. Structured test는 exhaustive test에서 불필요한 test pattern을 삭제하여 test하는 방법이다. 이 방법은 ATPG (automatic test pattern generator) 에서 사용하기도 하는 방법이다. Functional test는 단지 회로의 기능수행이 되면 정상이라고 판단하는 방법이다. 이와는 달리 VLSI test는 특성에 따라 다음과 같이 분류할 수 있다. Ad-Hoc approach, structured approach, self/ built-in approach로 분류된다. Ad-Hoc 방식은 주어진 회로의 특성에 적합한 test 회로를 첨가하는 방법을 사용하므로 모든 회로에 공통적으로 적용할 수 없다. 그러나 설계자에게 제약 조건을 부과하지 않으므로 testability 실현에 부담이 없으며, 특별한 회로에 쉽게 test 할 수 있어 test 비용을 줄일 수 있다. 그러나 ATE (automatic test equipment)로 test 프로그램을 생성할 때 test pattern의 자동 생성에 어려운 단점이 있다. 분할 방법, test point 추가, bus architecture를 이용한 방법 등이 이 부류에 속한다. 이 방법은 그림 5에서는 분할 방식을 이용한 test 회로를 보였다.

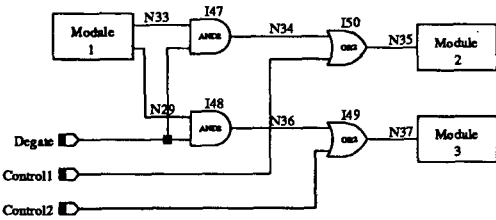


그림 5. 분할 방식을 이용한 test 회로

Structure 방식은 회로의 구조에 기초를 둔 체계적인 test 방법으로, 순차 회로의 test를 용이하게 수행하기 위하여 개발되었다. 순차 회로의 test를 조합 회로의 test로 변환하여 회로를 test하기 용이하도록 한다. 방법은 회로 내부에 scan register를 첨가하여 회로의 초

기값을 설정하거나 수행 결과를 관측할 수 있도록 하는 것이다. 또한 이 방식은 IC 내부 뿐 아니라 boundary scan 방법을 사용하여 PCB 에서 IC 상호간의 연결 상태까지도 test 할 수 있다. Scan method, random access scan method, LSSD method 등이 있다. 특히 이 방식은 자동 test pattern 생성에 효과적이다. 그림 6에서 structured test의 기본 구조를 보였다. 그림에서와 같이 조합 논리 회로에 기억 회로가 있다. 기억 회로는 조합 논리 회로의 출력을 기억하고 있어서 외부에서 그 결과를 쉽게 알 수 있다. 그림 7 및 그림 8에서는 double-latch LSSD를 보였다.

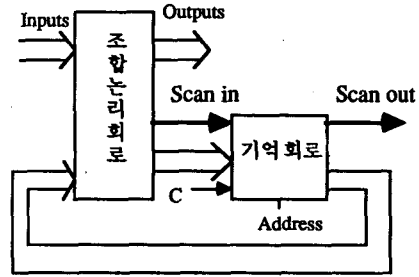


그림 6. Scan method의 구성도

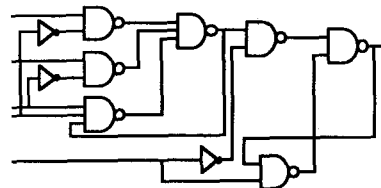


그림 7. SRL의 논리 회로도

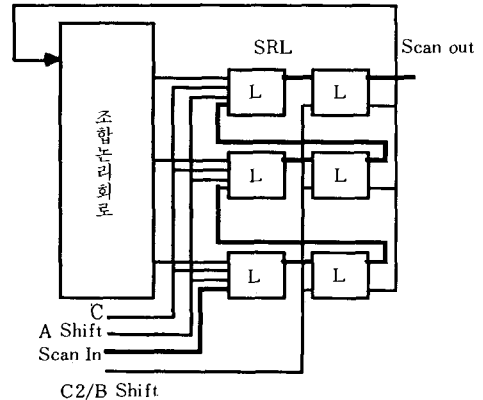


그림 8. Double-latch LSSD의 논리 회로도

한편 IC의 집적도가 높아지므로 많은 양의 test pattern이 요구되므로 self/built-in test는 IC 내부에서 자체적으로 오류를 검사하거나 test 결과를 줄여주는 설계 방법을 통하여 test pattern을 현저히 감소하도록 하는 설계 방식이다. Signature analysis 방법과 BILBO (built-in logic block observer)를 이용한 방법이 있다. Signature analysis 방법은 회로의 출력 또는 test point에서의 결과를 아주 작은 bit pattern으로 압축하여 볼 수 있는 특징으로 인하여 실제 test에서 많이 사용되고 있다. BILBO는 IC를 자체 test 하고자 할 때 사용하는 방법으로 하나의 회로로 여러가지의 기능을 수행한다. 정상 동작 상태에서는 단순 래치로, test 동작 상태에서는 LFSR; scan path, reset 등의 기능을 수행할 수 있다. 그림 9에서는 BILBO를 이용하여 test하는 기본 기능을 보였다.

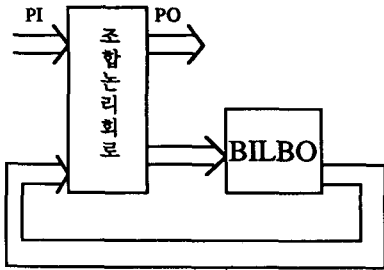


그림 9. BILBO를 이용한 test 회로의 기본 구조

LSI 및 VLSI를 test하기 위해서는 회로 자체가 다음 2가지 사항을 만족해야 한다. 하나는 제어 가능성 (controllability)이고 다른 하나는 관측 가능성 (observability)이다. 즉, test를 하기 위해서는 회로를 외부에서 제어 할 수 있어야 하며, 또한 이에 대한 결과(출력)를 관측 할 수 있어야 한다.

VI. 새로운 설계 방법

그림 10에서와 같이 1990년대 중반기에 설계되는 chip의 집적도는 1M gate 수준을 육박하게 될 것이다. 따라서 schematic capture를 이용하여 logic diagram을 그려서 F/E 설계를 하는 것은 무의미하다. 즉, time-to-market이 강조되어 schematic capture를 통한 F/E 설계를 하는 방식은 사라질 것이다. 그 대신에 logic synthesis tool이 널리 사용될 것이다. 또한 2000년대

초기에는 설계 규모가 10M gate 수준에 이르러서 behavioral synthesis tool이 등장하게 될 것이다. 상위 수준 합성 과정은 그림 11에서와 같다. 설계할 system의 사양이 결정되면 HDL (hardware description language)을 이용하여 논리 기술을 한다. 그리고 HDL simulator를 이용하여 HDL을 검증 및 수정한다. 검증이 끝난 HDL은 logic synthesis tool에 입력된다. Logic synthesis tool에서 HDL이 특정 technology와 결합되어 논리 합성을 한다. 그 결과를 B/E tool에 전달한다. 논리 합성에서는 IC의 면적과 속도에 대한 trade-off를 잘 결정하여야 한다.

상위 수준의 합성의 장점은 HDL로 기술된 설계가 특정 technology와는 무관하게 독립적이라는 것이다. 이는 최초 완성된 설계는 새로운 technology만 기술되면 재합성을 통하여 설계를 할 수 있다. 최근의 경향은 technology에 독립된 설계를 추구하여 technology 변화에 따른 재 설계과정을 줄이고자 한다. 상위 수준 합성

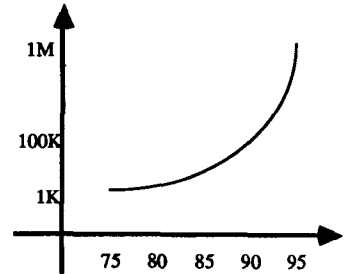


그림 10. 집적도의 증가 그래프

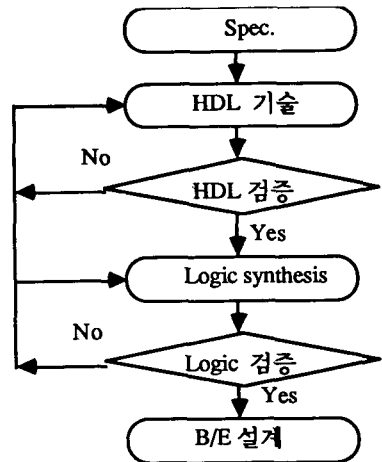


그림 11. High level synthesis flow

의 또다른 장점은 설계 결과가 숙련자와 초심자가 비슷하다는 것이다. 기존의 설계 방식에서는 숙련자의 설계 know-how가 많은 역할을 하였다. 그러나 논리 합성 tool에서는 이와같은 know-how를 rule base로 하여 합성하므로 설계자의 숙련도에 무관한 설계 결과가 나온다. 이 외에도 설계 과정이 빠른등의 여러가지 장점으로 인하여 상위 수준 합성을 이용한 설계는 ASIC 설계의 새로운 흐름이 되었다.

VII. ASIC CAD

ASIC 설계에 있어서 CAD tool의 역할은 매우 크다. 과거에는 turn key 방식의 일관된 CAD 환경을 제공하였으나, ASIC 설계가 복잡해짐에 따라서 CAD tool도 유연한 방향으로 변화되었다. 최근의 경향은 CAD tool 개개의 기능보다도 전체적인 CAD 환경구축이 더 중요시되고 있다. 이러한 변화는 tool, platform, technology 및 tool vendor에 독립된 CAD 환경 구축을 도모하는 것이다. 이에 반하여 CAD tool 자체는 점차 표준화를 추구하고 있다. 표준화된 입출력을 사용하고, 표준화된 data format을 유지하며, 표준화된 GUI (graphic user interface)를 사용하려 한다. 상위 수준 합성에서는 VHDL (very high speed IC HDL)을 표준 입력으로 사용하며, 설계 결과는 EDIF (electronic design interchange format)를 표준으로 출력한다. 또한 B/E 설계 tool에서는 GDSII가 이미 업계 표준 입출력으로 되어 있다. 이 외에도 X-11의 motif 환경을 표준 graphic 환경으로, UNIX를 표준 O.S로 채택하고 있다. 이외에도 CAD framework 개념의 도입은 CAD 환경의 독립화와 CAD tool의 표준화를 용이하게 하였다. 개개

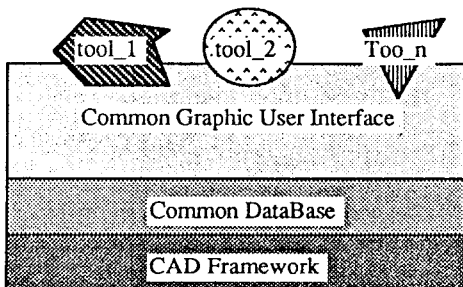


그림 12. CAD framework의 모형도


의 CAD tool은 tool vendor 와는 무관하게 CGUI (common GUI)를 통하여 접근 가능하고, CGUI는 CDB (common data base)를 접근하도록 한다. 이에 대하여 그림 12에서 보았다.

VIII. 결 론

ASIC 시장의 흐름은 급변하고 있다. 더불어서 설계 방식에도 많은 변화가 있었다. F/E에서는 상위 수준 합성을 통한 설계가 이루어지고 B/E에서는 prototype 용으로 FPGA가 널리 사용되고 있다. 설계자 또한 보다 system 수준에서의 설계에만 관심을 기울이게 되었다. 또한 CAD tool은 점차 technology 독립적이며 표준화를 추구하게 되었다. Framework이 등장하였고 CGUI를 채택하는 등의 변화가 있었다. 설계에 있어서도 공정 기술의 발전으로 delay 문제가 심각하게 되었고, 특히 test는 설계 초기 단계에서 부터 고려되어야 하는 중요한 사항이 되었다.

이러한 변화는 silicon compiler의 개발을 촉진시키게 되고, 설계자는 보다 알고리즘 개발에 집중하게 될 것이다.

參 考 文 獻

- [1] F.Lambardi, M. Sami, *Testing and Diagnosis of VLSI and ULSI*, Kluwer Academic Publishers, 1987.
- [2] B. Preas, M. Lorenzetti, *Physical Design Automation of VLSI Systems*, The Benjamin/Cummings Publishing, 1988.
- [3] E. Sternheim, R. Singh, Y. Trivedi, *Digital Design with Verilog HDL*, Automata Publishing Co., 1990.
- [4] OPUS Manual, Cadence, 1992.
- [5] GDT Manual, Mentor Graphics, 1991.
- [6] "ASIC Synthesizer with Verilog HDL", COMPASS Design Automation, 1990.
- [7] ASIA Electronics Engineer, 1990.1-1992.4
- [8] Manual Script, Synopsys, 1992. 

筆 者 紹 介


金 春 慶

1974年 서울대학교 전기공학과 (학사)

1981年 Univ. of Iowa (석사)

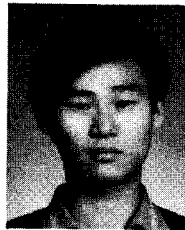
1984年 Univ. of Minnesota (박사)

1974年~1979年 국방과학연구소 연구원

1983年 Metheus Corporation 연구원

1984年~1989年 General Electric R&D Center 연구원

1989年 3月~현재 금성 일렉트론 이사


徐 仁 煥

1965年 2月 13日生

1988年 경북대학교 전자공학과 (학사)

1990年 경북대학교 (석사)

1990年~현재 금성 일렉트론 주임 연구원