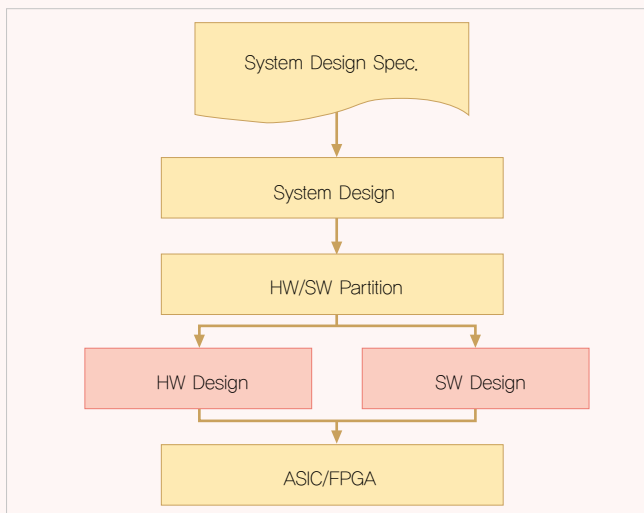


SoC H/W Design Flow

백영석 책임연구원_ETRI 모뎀Soc설계팀

서론

현재 우리 주위의 인기를 끄는 대부분의 제품들을 보면 휴대폰, 무선인터넷과 같은 통신관련 제품 이거나 MP3 플레이어, MD Player 등과 같은 신호처리를 수행하는 제품들이다. 그에 따라 SoC (System on a Chip) 설계분야에서도 이러한 디지털 통신 및 디지털 신호처리 분야의 설계가 주종을 이루고 있다. 이들 제품들은 다른 기능의 융합으로 점점 더 복잡해지고, 새로운 제품의 출시에 대한 주기가 점점 짧아지고 있다. 이러한 상황에서 관련 제품의 설계는 복잡한 사양을 좀더 짧은 시간 안에 이루어져야 한다는 요구를 수용할 수 있어야 한다. 이러한 SoC의 설계는 일반적으로 [그림 1]과 같은 방법으로 이루어진다.

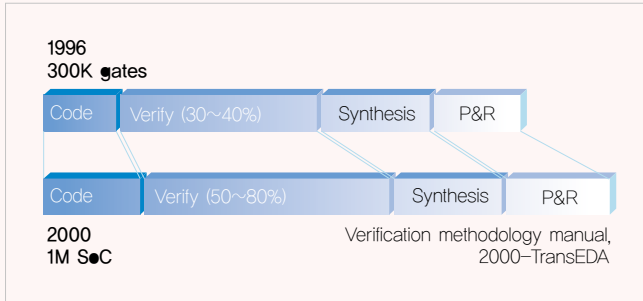


[그림 1] 개략적인 SoC 설계 Flow

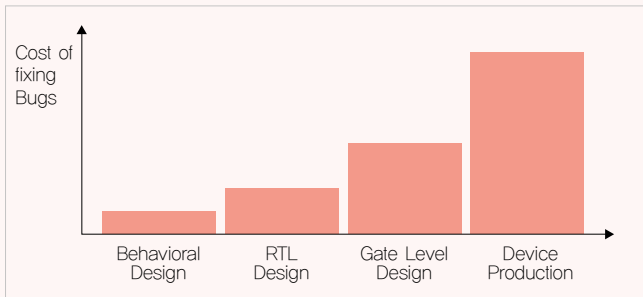
시스템에 대한 사양은 분석과정을 거쳐 시스템 설계언어로 표현된다. SoC를 위한 시스템 설계는 HW/SW 분할에 의하여 하드웨어 부분으로 구현하는 부분과 소프트웨어로 구현하는 부분으로 나뉘어진다. 이 분할 과정은 대부분 설계자의 경험에 의하여 이루어진다. 분할된 하드웨어 부분은 하드웨어 설계자에 의하여 설계가 이루어지고, 소프트웨어 부분은 소프트웨어 설계자에 의하여 설계 되어진다. 하드웨어와 소프트웨어의 연결관계는 인터페이스 설계를 통하여 나중에 하드웨어 인터페이스 부분과 device driver 부분으로 구성되어 각각 하드웨어와 소프트웨어에 통합된다. 시스템 설계, 하드웨어 설계, 소프트웨어 설계 각 단계에서 검증이 이루어지며, 하드웨어와 소프트웨어로 분할 된 후부터는 하드웨어/소프트웨어 혼합 시뮬레이션을 통하여 검증이 이루어지게 된다. 하드웨어 부분은 FPGA나 ASIC으로 구현되며, 소프트웨어는 내장된 프로세서 내에 탑재되어 수행되게 된다. [그림 2]에서 나타난 것과 같이 TransEDA에서의 발표자료를 보면 30만 게이트 급의 설계에 소요되는 비용과 100만 게이트 급의 설계와 비교하면 설계 코드의 작성은 약간 늘어났으나 이를 검증하는 시간은 거의 두배 정도 증가함을 알 수 있다. 그리고, 현재의 수백만 게이트 급에서 검증이 차지하는 비율은 전체 설계시간의 70%~80% 가 된다고 한다. 이는 디지털 설계에서 검증이 SoC 설계의 주요한 부분을 차지하고 있음을 알 수 있다. 이러한 이유로, 외국 설계 회사나 주요 설계 회사에서는 검증전문 엔지니어를 따로 두고 검증만을 담당하게 하고 있다.

또한 [그림 3]의 TransEDA 자료에 의하면 설계 검증에서 문제를 발견하고 수정하는데 걸리는 시간은 아래 그림과 같이 하위의 수준으로 갈수록 기하급수적으로 늘어나게 된다. 따라서 가능한 설계의 초기단계에서 검증이 충분히 이루어져야 한다. 결과적으로 상위수준설계 (Behavioral Design)와 RTL 설계에서 완벽한 검증이 이루어져야 전체 설계시간을 단축할 수 있다.

본 고에서는 하드웨어 설계에 이용되는 설계 흐름에 대하여 기술하고, 최근에 다시 활발하게 논의 되고 있는 상위수준 합성기에 대하여 간략히 설명하며 발표된 것들 중 대표적인 것에 대해서는 좀 더 자세히 기술하였다.



[그림 2] 설계규모에 따른 설계시간 비율



[그림 3] 각 수준별 bug 정정에 따른 설계 비용

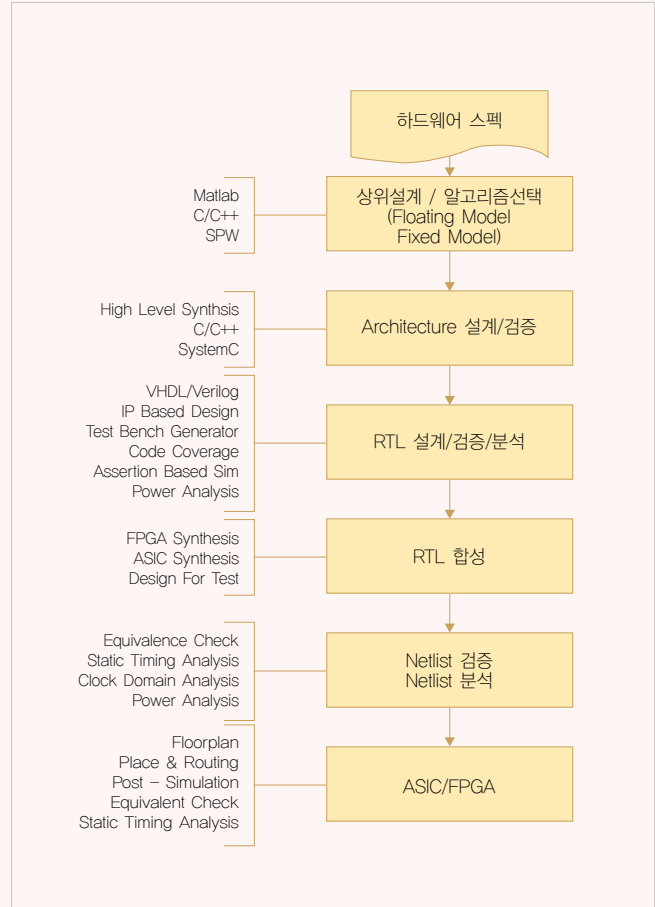
본론

| 일반적인 하드웨어 설계 방법 |

일반적인 하드웨어 설계 방법은 [그림 4]와 같다. 알고리즘 수준의 하드웨어 사양은 부동소수점 모델과 고정소수점 모델을 이용하여 검증이 이루어지며, 이로부터 구조설계와 RTL 설계가 이루어진다. RTL 설계는 다양한 설계검증 툴을 이용하여 충분한 검증을 수행하고 합성과정을 거쳐 게이트 수준의 설계로 변환된다. 게이트 수준에서 또한 다양한 검증과 분석을 수행하고, 추후 배치와 배선을 통하여 FPGA 나 ASIC으로 구현된다.

① 부동소수점 모델과 고정소수점 모델

설계자는 하드웨어 사양에 대한 분석을 통하여 사양의 동작에 대하여 정확히 이해하고 설계 성능의 지표, 전력소모 등을 추출하거나 결정한다. 이어서 상위 모델인 부동소수점 모델을 구현하게 된다. 이 부동소수점 모델은 다양한 알고리즘을 사용하여 주어진 성능지표의 만족여부를 판별하여 조건에 맞는 알고리즘을 선택한다. 부동소수점 모델은 이상적인 신호 값을 사용하므로 실제의 하드웨어에서 사용할 수 있는 신호 값을 이용하는 고정소수점 모델로 변환한다. 이때 선택된 알고리즘의 다양한 부분에 대한 시뮬레이션을 통하여 주어진 성능지표를 만족하며 가능한 한 적은 비트 수를 사용하는 모델을 찾는다. 이는 비트 수의 크기가 하드웨어의 크기를 좌우하기



[그림 4] 일반적인 하드웨어 설계 방법

때문이다. 고정소수점 모델은 다시 구조가 적용된 고정소수점 모델로 변환된다. 일반적으로, 부동소수점 모델과 고정소수점 모델은 서로 모델을 공유하며 개발할 수 있다. 그러나 구조적용 고정소수점 모델은 모델의 표현에 하드웨어 구조 정보가 같이 추가되므로 새로이 모델링 되어야 한다. 잘 설계된 구조적용 고정소수점 모델은 추후의 RTL 설계에 대한 reference 모델로 사용 된다.

② RTL 설계와 검증

RTL 설계자는 구조적용 고정소수점 모델을 기준으로 주요 알고리즘의 RTL (Register Transfer Level) 설계를 수행한다. RTL설계에 사용되는 언어는 표준화된 언어인 VHDL (VHSIC Hardware Description Language) 와 Verilog가 주로 사용되고 있다. SystemC 언어의 사용도 가능하나 이 언어를 지원하는 합성기가 다양하지 않아 아직 RTL 설계에서 주력 언어로 부상하지 못하고 있다. 최근에는 VHDL언어에 비해 Verilog 언어가 가지는 단점이 많이 보완한 SystemVerilog 언어가 Verilog의 차기 버전으로 나와 주목을 끌고 있다. 이 언어는 자체적으로 테스트벤치 자동생성기능 과

Assertion-based 시뮬레이션을 수행하는 기능을 포함하고 있다.

RTL 설계 검증은 주로 시뮬레이션을 통하여 이루어진다. 적당한 규모의 설계는 시뮬레이션을 이용하여 충분히 RTL 검증을 수행할 수 있다. 그러나 수백만 게이트급의 통합된 시스템에 대한 RTL 모델을 검증할 경우 시뮬레이션만을 이용하면 상당한 시간이 걸린다. 이러한 문제는 FPAG같은 하드웨어를 탑재한 시뮬레이션 가속기를 이용하면 시뮬레이션 속도를 수십 배 정도 높일 수 있다. 이는 RTL 설계를 가속기내의 FPGA에 맵핑하고 1MHz 정도 내외의 clock을 이용하여 시뮬레이션을 수행함으로써 가속기능을 수행한다. 이때, 설계자가 원하는 신호의 정보는 모두 탐색 가능하다. 시뮬레이션을 이용한 검증은 설계자가 수행 결과로 얻어진 신호 파형들을 일일이 분석하여 오류를 찾아 내어야 하므로 복잡한 동작이나 긴 시간의 시뮬레이션을 수행하는 경우 오류 탐색이 쉽지는 않다.

최근 수년 전부터 assertion 기능을 이용한 시뮬레이션 기법이 사용되기 시작하여 효과를 보고 있다. Assertion 기능은 초기에 VHDL 언어 내에서 지원하는 구문으로 조건 문 안에 기술된 조건이 true 인지를 검사하여 아닌 경우 원하는 정보를 출력하는 기능이다. Assertion based 시뮬레이션 기법은 이와 유사한 개념으로 보다 발전된 property 기술 방법으로 다양한 검증 조건을 기술하고 검사할 수 있게 한다. 설계에서 반드시 만족해야 하는 조건을 기술하고 수행결과 나온 위반사항을 점검함으로써 쉽게 설계 오류를 찾는 것이 가능하다. 이 property는 formal verification 방법에서 유래된 것으로 assertion based 시뮬레이션방법이 formal verification과 시뮬레이션 기법과 결합하여 사용되어서 semiformal 기법이라고도 이야기 한다.

시뮬레이션을 이용하여 검증하는 방법은 설계자가 부여한 테스트 벤치에 대하여만 검증이 이루어진다. 이것은 사용하는 테스트 벤치의 질이 설계 검증에 매우 중요함을 의미한다. 테스트 벤치 자동 생성 툴은 설계자가 쉽게 테스트 벤치를 생성할 수 있도록 도와준다. 한편, code coverage 툴은 테스트 벤치가 설계의 코드의 어느 부분을 검증하는지, 전체 설계에서 해당 테스트벤치로 어느 정도 검증이 되었는지를 분석하여 준다. Code coverage가 제공하는 측정 방법에는 statement coverage, branch coverage, path coverage, expression coverage, toggle coverage 등이 있으며, 이 가운데 statement coverage(또는 line coverage)와 branch coverage는 기본적으로 설계자가 의도적으로 기술한 redundant code를 제외하고 100%의 coverage를 갖도록 검증 되어야 한다. 이러한 테스트벤치 생성기와 code coverage 툴을 이용하여 RTL 설계의 검증의 질을 높일 수 있다. RTL에서 수행한 검증은 gate level 이하에서 이루어지는 검증에 비하여 매우 중요하다. 상위에서 검증이 잘 수행되면 하위의 검증시간을 대폭 줄일 수 있다. 따라서 가능한 한 RTL에서 완벽한 검증이 이루어져야 한다.

최근에는 formal verification의 한 분야인 model checking 기법을 RTL 검증에 이용하는 방법이 나타났다. 이는 설계 사양을 property를 이

용하여 기술하고 수학적 증명 방법을 이용하여 RTL 설계를 분석하는 방법이다. 이 방법은 테스트 벤치를 사용하지 않아 static 검증방법에 속한다.

③ 전력 예측기 및 전력 합성기

이동환경에서 사용되는 제품의 설계는 저전력 설계가 매우 중요하다. RTL에서 각 블록 내에 기술된 하드웨어 기술 표현을 분석하여 gated clock 같은 기법을 이용하여 저전력 표현으로 변환할 수 있는 정보를 제공하거나 변환시켜주는 소비전력 예측기(Power Estimator) 또는 소비전력 합성기(Power Synthesizer)가 있다. 소비전력 예측기는 RTL 코드를 분석하여 해당설계가 소비하는 전력의 양을 예측하여 준다. 뿐만 아니라 각 코드가운데 저전력 설계가 가능한 표현으로 변환이 가능한 것은 이에 대한 report를 제시하여 줌으로써 설계자가 이를 적용하도록 하는 기능이 있다. 동적인 소비전력은 신호의 변환에 크게 좌우된다. 즉, 신호선 가운데 항상 변환이 이루어지고 많은 소자와 연결이 되어있는 clock에 의한 동적 소비 전력 줄이는 것이 저전력 설계의 중요한 점이다.

전력 합성기는 RTL 코드를 합성할 때 gated clock 기법을 이용한다. 예를 들면 조건 A에 flip-flop이 동작하여 값이 갱신되는 경우 조건 A를 clock이 동작하는 조건(조건 A와 clock을 AND gate로 묶음)으로 gated clock의 형태로 변환하여 flip-flop이 항상 clock의 변환에 대하여 동작하지 않고 조건 A에만 동작하도록 변환시킨다. 이 기법은 동적 소비전력(dynamic power)을 효과적으로 줄일 수 있다. 블록 별 소비전력의 분석 결과를 통하여 TDD (Time Division Duplex)와 같이 정하여진 시간에 전송동작을 수행하고, 다른 시간에 수신동작을 수행하는 블록은 시간단위로 소비 전력의 분포가 차이가 심하게 나타난다. 이러한 블록은 gated clock 등을 이용하여 각 블록들을 해당하는 시간 동안만 동작하도록 변경하면 clock에 의한 동적인 소비전력은 크게 줄일 수 있다. 최근까지의 저전력 설계 기법에서는 동적인 소비전력이 정적인 소비전력에 비해 전체 소비전력의 많은 부분을 차지하기 때문에 동적인 소비전력을 줄이는 노력을 많이 하였으나, 90nm 이하의 technology에서는 정적인 소비전력이 오히려 중요한 요소가 된다. 따라서, 앞으로의 저전력 설계에서도 voltage threshold를 다르게 하거나, 인가되는 voltage를 바꾸거나 단락시키는 기법 등을 이용하여 정적인 소비전력의 양을 줄이는 방법이 더 많은 노력을 기울여야 할 것이다.

④ 게이트 수준 설계와 검증

RTL 설계와 검증이 끝나면 ASIC 논리합성기 또는 FPGA 논리합성기를 이용하여 게이트 수준의 net-list로 변환한다. 이 net-list는 합성 시에 설계자에 의하여 인가된 clock의 동작 주파수와 technology에 맞추어 설계된다. 게이트 수준에서의 검증은 시뮬레이션을 통하여 동작검증과 timing 검증을 수행한다. 그러나 일반적으로 수백만 게이트급의 설계에서 게이트 레벨의 시뮬레이션 기간은 수일 ~ 수주가 소요되게 된다. 하지만 효과적인

방법으로 알려져있는 formal verification 방법 중의 하나인 EC (Equivalence Checking) 와 STA (Static Timing Analysis) 라는 static 검증기법을 사용하면 하루 내에 검증을 수행할 수 있다. EC는 두개의 설계가 서로 동일할가를 수학적 방법을 이용하여 검증하는 것으로 이 방법을 사용하기 위해서는 오류 없는 reference가 되는 설계가 존재 하여야 한다. EC를 사용하는 설계흐름에서는 RTL 설계의 검증을 충분히 하여 RTL 설계가 오류가 없는 golden RTL 이라고 가정하고 이 방법을 수행한다. 게이트 수준의 검증은 기능과 지연시간의 검증이 다 이루어져야 한다. EC는 게이트 수준의 설계가 golden RTL 설계와 기능이 같은가를 판별하는 것을 담당하고, 지연시간 검증에 대한 것은 STA 가 담당한다. STA는 주어진 clock의 동작시간 내에 모든 조합회로(Combinational Circuit)가 안정된 값을 전달하는 가를 계산하고 위반된 경로의 정보를 알려준다. 즉, Clock의 동작 시간을 초과하는 경로에 대한 정보를 설계자에게 알려주며, 설계자는 이 지연시간의 위반이 일어나는 부분을 분석하고 이를 수정할 수 있도록 설계를 수정하여야 한다.

EC, STA와 같은 정적 검증기법은 테스트 벤치의 사용이 필요 없으며, 수행시간이 수백만 게이트/day로 매우 빨라 설계자들의 사용이 계속 증가 되는 추세이다.

⑤ Floorplan, 배치, 배선

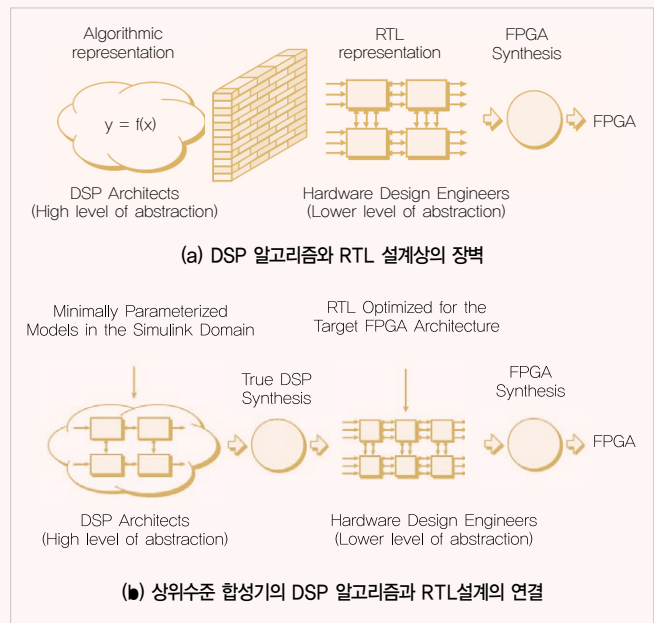
합성된 net-list는 ASIC이나 FPGA로 구현되기 위하여 배치, 배선된다. 이 배치 배선을 효과적으로 수행하기 위하여 floorplanning을 하게 되며 매크로 셀이나 표준 셀들의 위치를 지정하고 분석하여 배치 배선이 완료되도록 한다. Technology가 점점 낮아지면서 배선에 대한 지연시간이 전체 지연시간에 주요한 부분을 차지하게 되면서 지연시간을 고려한 배치기법이 이용되고있다. 최근에는 배선의 지연시간이 주요 문제가 되면서 RTL 합성 시에 배치문제를 고려하는 RTL floorplanning 기법이나 배치, 배선 시에 부분적인 논리합성을 수행하는 기법도 이용되고 있다. 이는 배치 배선 시에 100% 배선이 이루어지지 않거나 지연시간이 주어진 시간을 만족하지 않으면 아무 의미가 없는 chip이 되어 버리기 때문이다. 앞으로 점점 RTL 합성과 floorplanning의 두가지 문제는 밀접하게 연관되어 설계가 이루어져야 만이 양질의 부품이 생성될 수 있을 것이다.

| 상위 수준 합성기 |

상위수준 합성기는 구현하고자 하는 알고리즘에 대한 기술을 입력으로 하여 설계자가 설정하는 요소에 따라 다양한 architecture를 생성하고 이에 대한 RTL 코드로 변환하여주는 설계 툴이다. RTL 설계방법에서 상위수준합성으로 설계 방법의 변화는 기존의 스키매틱을 이용한 설계에서 하드웨어 기술언어를 이용한 RTL 설계로의 설계방법이 바뀌는 것에 해당하는 매우 큰 설계방법의 변화에 해당한다. 그러나, 지난 10여년 전부터 계속적으로 Cathedral, Behavioral Compiler, Visual Architect, Mone 등의

다양한 상위 수준 합성기에 대한 툴이 나왔지만, 대부분이 설계 방법을 RTL 설계에서 상위수준 설계로 끌어 올리지 못하고 사라지거나 명맥만 유지하고 있는 상태이다. 이것 들은 입력언어로 VHDL 같은 하드웨어기술 언어를 사용하였는데, 이는 사용상의 제한 점이 많아 RTL 설계자가 이러한 언어를 이용하여 알고리즘을 기술하기에 그다지 적합하지 않았다. 그런데 이러한 상위수준 합성분야에 2004년 무렵부터 다시 활발하게 새로운 툴들이 나타나고 있다. Mentor Graphics사의 Catapult C, Forte Design사의 Cynthesizer, Synplicity사의 SynplifyDSP, AccelChip사의 Accelchip DSP Synthesizer 등이 최근에 발표되었다. 이들이 사용하는 언어는 Catapult C는 C언어를 이용하며, Cynthesizer는 SystemC를 이용하고, SynplifyDSP는 Matlab의 Simulink를 이용하며, Accelchip DSP Synthesizer는 Matlab의 m언어를 이용한다. 이들 언어의 공통점은 알고리즘 전용 언어라는 것이다. 이 가운데 인상적인 것은 신호처리나 통신 분야의 모델과 시뮬레이션을 위하여 많이 사용하는 Matlab을 이용하는 것이다. Matlab에는 다양한 관련 함수(Toolbox)와 block(Blockset)들이 미리 정의 되어 있으며 설계자는 쉽게 이것 들을 이용할 수 있다.

SynplifyDSP는 Simulink 상의 Blockset list에 SynplifyDSP가 제공하는 RTL로 합성 가능한 Blockset을 추가한다. 설계자는 이 Blockset을 이용하여 구현하고자 하는 알고리즘을 표현하고 상위수준 합성을 수행한다. Blockset은 일종의 상위수준의 IP 또는 알고리즘 IP라고 할 수 있다. 즉 RTL 수준에서 제공하는 기존의 IP는 하나의 구조로 정하여진 상태로 설계자가 원하는 latency나 성능 등이 달라져도 바꿀 수 없는 반면, 상위수준의



[그림 5] SynplifyDSP 상위 수준 합성기의 역할

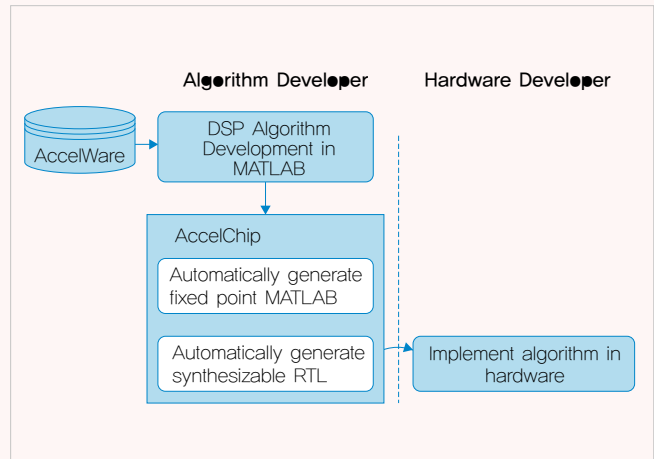
IP는 합성과정에서 이들의 값들을 변화시킬 수 있다. 현재 SynplifyDSP에서 제공하는 합성 가능한 blockset은 간단한 math 관련 blockset과 Signal Processing Blockset 이다. 제공하는 Block set들은 계속 추가될 예정이다. SynplifyDSP는 현재 FPGA설계 흐름만 설정되어 있으며, 추후 ASIC 설계흐름도 지원할 예정이다. [그림 5]는 Synplicity 사에서 발표한 SynplifyDSP의 역할을 보여주고 있다.

기존에는 DSP 알고리즘 설계를 마친 후 다시 RTL 설계를 손으로 재설계하여야 하였지만, 상위수준 합성기를 사용하는 방법에서는 SynplifyDSP를 통하여 DSP 알고리즘이 검증되면 직접 상위수준 합성과정을 거쳐 RTL 코드를 생성하고 FPGA까지 구현할 수 있어 설계 비용을 대폭 줄일 수 있게된다. SynplifyDSP가 Simulink의 라이브러리인 blockset 형태로 제공하는 상위수준의 IP는 architecture에 종속되어 있지 않으므로 설계자가 원하는 quantization 값을 설정하면 해당 값을 파라미터 형태로 전달하여 그에 해당하는 고정 소수점 IP를 생성할 수 있다. 이러한 IP들을 이용하여 설계된 상위수준 설계는 스케줄링 (folding), 자원할당, retiming, multichannelization, architecture 선택 등을 거쳐 설계자가 원하는 RTL 설계를 생성하게 된다. 스케줄링은 하드웨어 자원을 공유하기 위하여 수행하는 연산의 수행 시간을 정하는 것이며, 자원할당은 이 연산에 하드웨어를 연결시키는 것이다. Retiming은 pipelined 설계가 가능하도록 register를 추가하여 하드웨어의 수행시간을 균형있게 조절하는 것이다. Multichannelization 기법은 여러 신호가 같은 기능을 수행하는 동작을 하나의 신호에 대하여 구현하고 이를 다른 신호에도 적용하는 기법이다. 현재까지는 이 툴은 FPGA 응용 설계에만 적용되고 있지만 조만간 ASIC 설계도 지원할 것으로 예상되고 있다.

AccelChip은 Matlab 응용 프로그램에서 사용하는 m 언어를 입력으로 사용한다. M 언어는 추상화(Abstraction) 수준이 높아 알고리즘을 기술하기에 적합하며 다양한 알고리즘의 함수가 내장된 toolbox 라는 라이브러리를 통하여 DSP 관련 모델이나 통신관련 모델을 쉽게 만들 수 있다. 한 예로 FFT 기능의 함수는 단 한 줄의 문장으로도 모델링이 가능하다. AccelChip은 m 언어 내에 있는 많은 내장 함수와 Toolbox 내의 함수들을 AccelWare 라는 라이브러리로 제공하고 있다. AccelWare 라이브러리는 순수한 합성 가능한 파라미터화된 고정소수점 모델 형태의 m언어로 되어 있다. 이는 AccelWare가 상위수준 합성기를 이용하여 다양한 구조를 생성할 수 있는 상위수준 IP (알고리즘 IP)임을 의미한다. 제공되는 IP는 간단한 math 및 matrix 처리 관련함수에서부터 signal processing 함수, 그리고, 리드솔로몬 코덱, 콘볼루션 코덱 등과 같은 매우 큰 통신용 함수에 대한 IP까지 제공하고 있다.

AccelChip에서는 설계가 정의된 m 언어파일과 이를 이용하여 시뮬레이션하는 script 파일로 구분하여 설계하도록 권고한다. 설계 정의 m 언어 파일과 script m 언어 파일을 이용하여 부동소수점 시뮬레이션을 통하여 검증을 수행한다. 검증된 설계는 AccelChip 내에서 자동으로 고정소수점 모델로 변환된다. 설계자는 자동 생성된 입출력 또는 변수의 비트 수를 조

정하며 설계 최적화 작업을 수행한다. 설계가 완료된 고정소수점 모델은 상위수준 합성 기법을 이용하여 구조가 결정된 RTL 설계로 변환된다. 설계정의 m 언어파일로부터 합성 가능한 RTL 설계를 생성하고, script m 파일로부터 테스트벤치를 자동으로 생성한다. 생성된 RTL 설계는 VHDL 언어 또는 Verilog 언어 가운데 하나로 설정될 수 있다.



[그림 6] AccelChip의 설계 흐름도

RTL 설계 생성시 설계자는 FPGA 용과 ASIC 용 가운데 선택하게 된다. FPGA용은 FPGA사에서 제공하는 IP를 이용하여 RTL 코드를 생성하고, ASIC 용은 모든 부분의 코드를 생성한다. 이는 각 제품의 특성에 맞는 RTL 코드를 생성하기 위한 것이다. 이러한 RTL 코드는 기존의 하드웨어 설계 방법에 따라 검증과 합성을 통하여 FPGA 또는 ASIC으로 구현된다.

결론

본 고에서는 SoC 설계 방법 가운데 하드웨어 설계 방법의 전통적인 방법에 대하여 기술하였다. 그리고 최근에 RTL 에서 추상화가 한 단계 올라간 상위수준에서의 합성기에 대한 소개를 하고 대표적인 두 가지 합성기에 대해서는 좀더 자세히 기술하였다. 이러한 상위수준 합성기 들은 DSP나 통신 분야의 설계자가 검증에 널리 사용하는 matlab의 기술 표현을 입력으로 사용하여 설계자가 쉽게 사용할 수 있는 장점이 있다. 또한, 합성 가능한 상위수준 IP(라이브러리)를 제공하여 기존의 상위 수준 합성기가 제공하지 못한 강력한 설계능력을 제공하고 있다. 이러한 합성기 툴들이 아직은 초기 단계의 툴임을 감안하면 추후에 계속되는 IP들의 보강으로 저 설계 비용과 저 금보다 월등한 설계 능력을 제공할 것으로 예상되기 때문에 앞으로의 설계 흐름 상에서는 상위수준 합성기가 당연히 포함될 것으로 생각된다. ☺