특 집

DSP의 기술동향

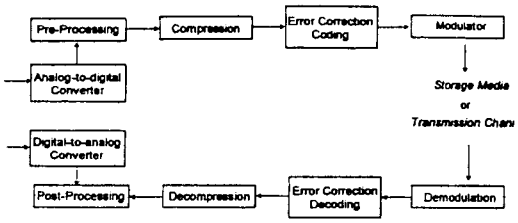
李 芳 遠

三星電子 半導體部門

I. 서 론

90년에 접어들면서 浮刻되기 시작한 multimedia의 열풍이 이미 모든 생활 전반에 큰 영향을 미치고 있다. 이와 같은 변화는 크게 사무실과 가정에 각기 다른 양상을 띠고 있다. 사무실의 환경은 정보전달과 배급을 요구하는 통신 중심의 성격인 반면, 가정의 환경은 entertainment와 edutainment를 위한 영상과 음향이 중심이 된다. 사무실에서 사용되는 통신 중심의 multimedia는 정보를 쉽게 찾아 내게 하는 Client-Server 중심의 정보추출과 LAN을 根幹으로 하는 전자우편시스템, Internet을 이용한 Web Browser 등으로 전개되고 있다. 또한, 공간의 이동 便宜性을 높이기 위한 무선통신 기술의 발전과 정보의 전달속도 및 규격화를 통한 분배성을 증대시키는 초고속 정보망의 설치가 변화의 속도를 가속화 시킬 전망이다. 이에 反해, 가정에서의 Multimedia환경은 크게 Personal Computer(PC)의 급격한 보급으로 변화가 가속화 되어 PC를 중심으로 이루어지고 있다. PC환경에서의 multimedia에서 개발되고 있는 것으로는, 손쉬운 기능추가 및 성능 개량을 쉽게 하기 위해 hardware 측면에서는 Plug-and-Play 기술이 있고, CPU와 DSP processor를 이용하여 주요 기능을 Software로 구현하는 기술들이 있다. 이와 같은 multimedia기술의 발달은 종래의 기술발전과 변화의 정도와 성격에 근본적인 차이가 있다. 이는 algorithm과 architecture 등으로 대변되는 digital signal processing기술이 그 변화의 중심에 있기 때문이라고 하겠다.

〈그림 1〉이 일반적인 映像과 音響을 중심으로 한 Multimedia시스템의 개념도이다. 映像信號와 音響信號는 前처리과정을 거친 다음 신호 압축의 단계로 들어간다. 여기서 前처리 과정은 사람에는 필요없는 신호 성분을 제거하는 기능과 신호 압축의 편리성을 높이는 역할을 한다. 압축에서 사용되는 방법은 일반적으로 다른 器機와의 共用化를 위해 국제적으로 표준으로 정해진 방법을 사용한다. 압축된 映像/音響 신호는 전송이나 기억매체에



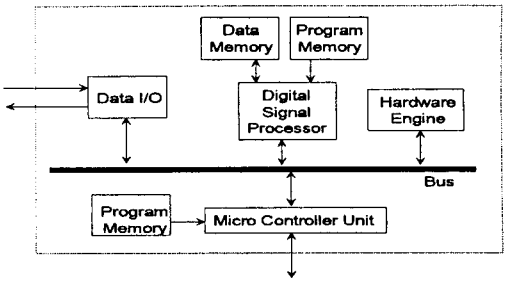
〈그림 1〉 Audio-Video 중심의 Multimedia System

기록을 위해 符號화의 단계로 들어간다. 이 단계에서는 전송망에서의 신호왜곡/손실이나 기억매체의 신호 손실에 무관해 지도록 원래의 신호에 오류정정을 위한 신호를 추가시킨다. 수신단이나 player에서는 위의 逆과정을 거쳐 이루어진다. 이와 같은 신호처리는 DSP기술을 바탕으로 하는 것이다. 여기서, 前처리/後처리 과정과 압축/복원 과정을 Source coding/decoding과정이라 하고 符號화와 복호화 과정을 channel coding/decoding이라고 하고 각각은 다른 이론적인 성격을 갖는다.

이외의 DSP기술을 사용하는 분야로는, hard disk의 servo계, motor의 효율제어, 주변조건에 변화에 따른 적응 제어와 같은 제어분야와 신호 인식, 목표 식별과 같은 인식 분야 등이 있다. 이와 같은 DSP반도체의 보편화 추세는 모든 전자기기의 digital화와 새로 창조되는 응용을 위해서는 수식으로 표현되는 algorithm의 필요에 따르는 자연적인 것이라고 하겠다.

II. MCU, DSProcessor, DSProcessing-LSI

DSP와 종래의 MCU(Micro Controller Unit)의 근본적인 차이는 DSP는 數式 演算을 처리하도록 되어 있고, MCU는 control동작을 처리하게 된 것이다. 따라서, programming을 하기 위해서는 DSP는 數式으로 표현되는 algorithm이 필요한 反面, MCU는 조건문과 control이 주가 되는 flow-chart가 필요하게 된다. 그 결과 DSP는 응용에 따라 구조가 달라지게 되고, MCU는 汎用의 성격



〈그림 2〉 DSP를 사용한 器機의 내부 구조도

을 갖게 된다.

〈그림 2〉는 일반적인 DSP를 사용하는 器機에서의 반도체 구성도이다. DSP는 주로 신호처리를 맡고 MCU는 주변 器機와의 제어를 맡고 있다. 實時間 신호 입출력은 DSP로 연결되어 있어 신호 입출력에 의한 지연을 최소화 하고 있다. 그러나, 이와 같은 DSP와 MCU의 분명한 차이는 점점 없어지는 추세가 있다. 이는, 반도체 집적화 기술의 발달로 system-on-chip이 가능해 진 것과 DSP기술이 家電機器에 용용이 되면서 super-integration을 통해 가격을 절감해야 하기 때문이다. 그 결과, 개발에 難易度가 높은 固定 소숫점 DSP의 사용증가 및 DSP와 MCU를 복합화 하는 경향으로 진행되고 있다.

MCU와의 複合化 추세에서는, MCU에 수식연산에 필요한 기능 블럭을 추가시킨 것과 MCU에 별개의 DSP Core를 붙인 것이 있다. 前者는 MCU의 register를 중심으로 하는 architecture에 multiplier와 accumulator를 추가한 형태가 되며, 대표적인 제품으로 68HC16 80196, ARM processor, PA7100LC 等이 있다. 이것은 고급언어를 지원하는 compiler의 지원이 원활한 현재의 control 중심의 기능에 단순히 수식연산을 지원하는 몇 개의 instruction을 추가하여 DSP응용을 부가적으로 처리하게 하기 위함이다. 이런 단순한 기능 블럭의 추가는 수식연산이 많은 응용에서는 한계가 있게 마련이다. 이에 反해, DSP Core와 MCU를 동시에 내장한 형태는 단순히 종래의 별개의 반도체를 하나로 집적화한 형태로 programming은 각각 독립적으로 하고 두 개의 Processor간에 data

〈표 1〉 Processor의 구조 방식에 다른 比較

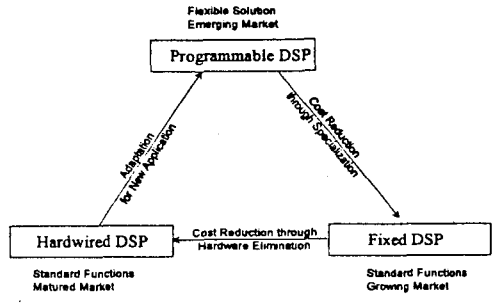
	Accumulator中心의 구조	Register中心의 구조
Stack의 구조	限定된 크기의 Hard-ware	Memory map 일부의 Software
명령어	-限定된 숫자 -수식 연산위주	-많은 숫자 -입출력 위주
Memory	-限定된 addressa-bel range -수 KByte의 data memory -수십 KByte의 pro-gram memory	-Data와 program memory가 통합 -수 Mega Byte이상의 addressabe ra-nge
Bus구조	Harvard	von Neumann
응 용	Em bedded DSPrec-essor	Micro Controller RISC Processor 高性能 DSProcessor

전달을 위한 부분을 추가한 구조를 갖고 있다. 이에 해당되는 제품으로 Zilog社의 Z86C94, Intel社의 K19이 있다.

〈표 1〉에 accumulator중심의 architecture와 register중심의 architecture의 차이점을 나타내었다. Accumulator중심의 구조에서는 數式 연산을 효율적으로 처리하도록 되어 있어 두 개의 data를 fetch하여 곱하고 合하는(혹은 減하는) MAC (multiplication-and-Accumulation) 기능이 한 개의 cycle에 처리한다. 따라서, 모든 data의 traffic이 accumulator로 집중되게 되어, 간단한 MCU기능인 address계산에는 적어도 3~4개의 cycle이 걸리게 되는 단점이 있다. 구체적으로 설명하면, address content를 accumulator에 fetch, offset address를 合한 다음, address register에 저장하는 3단계를 모두 accumulator를 거치기 때문이다. 이에 反해 register중심의 구조에서는 몇 개의 register를 address register로 사용하여 address register 숫자가 많고 register content를 수정하는 것도 보통은 1cycle내에서 처리하고 있다. 또, accumulator 중심의 구조는 data traffic이 accumulator에 집중되어 底電力化에는 근본적인 한계가 있게 된다. Texas Instruments社의 初期 DSP(32010, 320C20, 320C50 series)에서는 data traffic의 문제를 accumulator와 register를 혼합한 구조로 보완하였고, 제어와 신호 입출력을 위한 peripheral을 내장함으로서 범용성을 갖도록 하였다. 이같은 구조는 MCU에 익숙한 사용자에게 Texas Instruments社 DSP는 MCU와 유사한 환경에 빠른 수식 연산 능력을 갖고 있다는 인식을 주어 현재의 높은 시장 점유율을 갖게 된 것으로 보인다.

일반적으로 DSP라고 통칭하여 부르는 반도체를 명확하게 구분하면 2가지의 다른 형태가 있다. 하나는 digital signal processing LSI(DSP-LSI)이고, 또 다른 하나는 digital signal processor (DSProcessor)로 구분된다. 前者는 廣義로 해석하면 digital신호를 처리하는 모든 반도체를 통칭하는 것이지만 구분을 위해서는 신호처리에 필요한 algorithm을 hardwired한 logic으로 구현한 것이라고 할 수 있다. 後者는 數式 演算을 위한 특별한 기능 블럭(예를 들면 ALU, Multiplier 등)을 가지고 신호처리를 software로 구현하는 일종의 processor범주에 포함되는 것이 된다.

동일한 DSP기능을 DSP-LSI방식과 DSProcessor방식 중 어느 방식으로도 구현할 수 있다. 그 선택은 〈그림 3〉과 같은 시장의 상황과 신상품의 등장에 의해 시간대 별로 이루어진다. 새로운 개념의 제품이 등장할 때는 시장크기가 제한되고 구현될 기능의 변경이 자주 생기기 때문에 programmable DSProcessor형태로 개발이 된다. 시장이 점차로 성숙기에 들어 가게 되면 주요 기능이 확정이 되면 시장의 수요가 점차로 증가되고 가격에 대



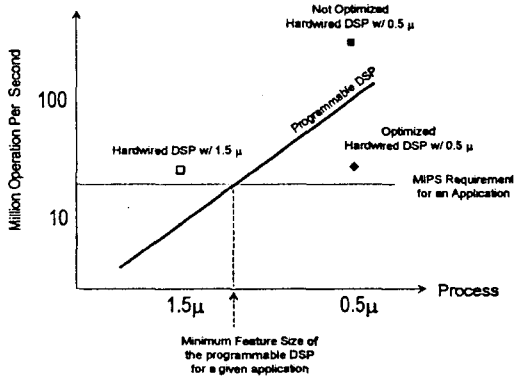
〈그림 3〉 DSP반도체의 발전 Model

한 경쟁이 요구된다. 이렇게 되면, fixed-function DSProcessor형태로 개발되게 된다. 그 이후, 제품이 완숙기에 들어 가면 새로운 추가 기능의 요구는 없어지고 가격만이 가장 중요한 요소로 남게 된다. 이렇게 되면, 모든 것이 hardwired된 DSP-LSI방식으로 개발된다. 이와 같은 상황에서 새로운 개념의 상품이 탄생되면 또 다시 신속한 제품개발이 가능한 programmable DSProcessor방식으로 되돌아오는 循環을 하게 된다.

DSP-LSI 방식과 DSProcessor 방식의 선정을 결정하는 또 다른 기술적인 요소로는 반도체 공정에 따르는 반도체 소자의 성능이 있다. DSProcessor 방식은 그 성능을 구성하는 주요 기능 블럭인 ALU와 multiplier의 속도가 실시간 응용에서 요구되는 속도를 따르지 못하면 사용 자체가 不可能하게 된다. 이에 反해, DSP-LSI는 요구되는 성능을 여러 개로 나누어 처리할 수 있도록 구조를 설계하여 주어진 반도체 공정에서 DSProcessor가 구현 불가능한 것을 구현할 수 있다.

또한 上位 반도체 공정을 적용하는 경우, DSProcessor 방식에서는 처리 성능이 향상되어 응용에서 요구되는 최소 성능 이상이 될 경우 여유분의 성능을 새로운 기능 추가에 사용할 수 있게 된다. 여유 성능이 너무 많은 경우에는, 다른 구조를 갖는 DSProcessor를 선택할 수 있다. DSP-LSI 방식의 경우에는 上位 반도체 공정을 선택하고 단순 layout만을 축소할 경우, 전체 회로에서의 설계 여유가 증가된다. 여기서의 설계 여유는 반도체 면적감소를 위한 가능성을 의미한다. 따라서, 上位 공정 선택에 의한 最適化 제품을 위해서는, DSProcessor 방식은 새로운 하위 구조를 가진 DSProceesor를 선택을 하게 되고 DSP-LSI 방식은 architecture단계부터 새로운 설계를 하여야 한다.

반도체 공정과 같은 맥락에서 각 방식의 동작 전압에 따른 성능 변화 특성이 방식을 선정하는 요소가 된다. MOS 공정으로 제작된 논리회로는 동작 전압의 감소에 따라 속도는 반비례한다. 따라서, 동작 전압의 변동에 따라 몇 개의 DSP블럭의 성능에 의해 좌우되는 DSProcessor에 비해 여러 개의 블럭으로 분산되어 있는 DSP-LSI가 동작전압



〈그림 4〉 반도체 공정에 따른 DSP설계 방식의 변화

의 변화에 덜 민감하다. 이와 같은 특성은 특정 설계 결과물을 전원 전압이 다른 응용으로의 재사용할 수 있는 면에서 DSP-LSI가 더 유리하다는 것을 보여주는 것이다. 〈그림 4〉의 경우는 전원 전압은 일정하다는 가정을 한 것이지만, 반도체 공정이 0.5micron 이하가 되면 최소선폭의 감소에 따라 전원 전압도 같이 감소하게 되어 上位 공정 적용에 의한 DSProcessor의 비례적인 성능 향상은 기대하기가 힘들어 진다. 따라서, 高性能 DSProceesor의 경우에는 여러 개의 DSProcessor를 병렬도 사용하여 응용에 필요한 연산 속도를 맞추고 있다. 이에 반해, DSP반도체를 사용하는 휴대용 기기에서는 전력소모를 줄이기 위해 낮은 전원 전압을 사용하고 있다. 이와 같은, 抵전력 응용에서는 DSP-LSI가 유리한 선택이 된다.

III. Host Signal Processing技術과 Coprocessor Signal Processing技術

RISC processor 성능의 급격한 발달로 DSProcessor의 필요성여부가 논란이 되고 있다. 이는 DSProcessor와 RISC processor는 대부분의 명령어가 간단하고 거의 1개의 cycle내에서 동작하는 유사성이 많고, 단순히 성능을 나타내는 parameter로 MIPS(Million Instruction Per Second) 수

치를 사용하기 때문이다. 이와 같은 관점에서 볼 때, 최근의 RISC processor가 수 백 MIPS 이상의 성능이 나옴에 따라 host processor를 이용한 DSP기능 실현을 시도하고 있다. 그러나, 이런 RISC processor의 MIPS수치는 측정하는 benchmark는 수식 연산이 主가 아니어서 DSProcessor의 MIPS와 같은 다른 의미를 갖고 있다. 이런 RISC processor의 수식 연산의 부족을 보완하기 위해 multiplier와 accumulator를 추가시키기도 하지만, 응용별 각기 다른 연산 방식을 요구하는 DSP기능을 범용으로 처리하는 데에는 한계를 보이고 있다.

좋은 예로서, Intel이 제안하고 있는 NSP(Native Signal Processing)기술이다. 이는 Intel의 Pentium Processor의 성능이 integer benchmark인 SPECInt92가 100Mega 이상이 됨에 따라 단순히 algorithm계산에서의 MIPS와 비교하여 Multimedia에 필요한 audio/video기능을 host processor에서 처리하겠다는 계획이였다(이 때문에 NSP를 host signal processing이라고 부르기도 한다). Intel社의 NSP기술은 추가 비용 없이 multimedia기능을 personal computer에 구현할 수 있다는 점에서 큰 반향을 불러 일으켰다. 이것을 기존의 DSProcessor업체로서 큰 위기로 받아들여져 그 실현 가능성 확인에 최선의 노력을 하고 있다. 현재까지 발표된 benchmark결과에 따르면 그 CPU는 interrupt를 중심으로 동작하는 실시간 응용에서는 효율성이 1/3로 떨어진다는 것이다. 구체적인 예로서, 음성압축의 de facto standard인 DSPG社의 TrueSpeech의 경우 Pentium 90MHz의 60%의 계산량을 차지하는 反面, Motorola社 24-bit 固定 소숫점 DSP인 56002로는 13.5MIPS를 차지한다는 것이다. 특히 MAC동작과 adaptive 제어를 위한 memory update가 主된 동작인 modem의 data pump 부분은 CPU로 구현하기에는 非效率的인 것으로 보고되었다. 따라서 최근에는 host processor이외에 별도의 DSProcessor를 동시에 사용하여 상호 보완을 하는 방향으로 움직이고 있다.

IV. DSProcessor의 現況

DSProcessor는 高速化, 高機能化, 高性能化의 경향과 low-end embedded화 경향이 兩極化되는 현상이 생기고 있다. 고성능 DSProcessor는 programmable하고 고속 신호를 처리하는 성능에 대한 요구와 top-down설계환경을 지원하기 위해 설계 tool에 대한 便宜性을 만족시키기 위한 것이다. Architecture측면에서는 register를 중심으로 하고 Harvard architecture에서 사용하는 dual bus를 채용하거나, 아예 von-Neumann architecture와 Harvard architecture를 혼용하여 multiple bus를 채용하기도 한다. 또한, data를 고속을 처리하기 위한 pipeline을 모든 engine에 사용하고, 여러 개의 engine을 병렬 배치하거나 아예 동일한 engine을 여러 개를 묶어서 vectorize하기도 한다. 이런 고속 DSProcessor의 발전은 이미 수 GOPS(giga operation per second)이상의 성능을 내고 있어, 가장 많은 演算 처리 능력을 요구하는 動 영상의 압축을 software로 처리하는 DSProcessor도 개발되고 있다. 예를 들면, Texas Instruments社의 TMS320C80는 동일한 DSProcessor를 여러 개 병렬로 구성하였고, Chromatic Research社의 VSP와 Philips社의 MediaDSP는 각기 다른 기능의 DSProcessor를 병렬로 처리한 일종의 VLIW (Very Long Instruction Word)와 유사한 구조를 갖고 있고, 三星電子와 Array Microsystems社가 공동 개발한 VideoFlow DSProcessor는 각기 다른 기능 engine간의 同期를 data availability로 하는 dataflow구조를 갖고 있다.

이에 反해, 家電器機에 사용되는 embedded DSProcessor는 固定 소숫점 형태로 accumulator 중심의 구조에, MCU에서 사용되는 peripheral을 갖고 있는 것이 보통이다. 이는 가격이 우선이 되는 용도에 맞도록 작은 die면적과 single chip과를 위해 선택된 결과이다. 작은 면적에 비해, 개발 tool의 한계로 인해 개발에 필요로 하는 노력과 시간은 상당히 많이 필요로 하는 문제점이 있다.

Accumulator를 위주로 하는 DSProcessor는

register를 중심으로 하는 MCU에 비해 고급언어 complier에 적합하지 못해 많은 한계가 있다. 예를 들어, 固定小數點 DSProcessor를 C언어로 programming한후 C-compiler를 사용하면 assembly 언어로 사용한 것에 비해 2배에서 10배 정도로 많은 명령어가 만들어 진다. 이를 보완하는 방법으로 DSProcessor내에 많은 memory를 내장시키거나, 浮動 소수점 구조의 DSProcessor를 여러 개 사용한 multi-processor구조가 있다. 그러나, 이런 방법 또한 低價格 용도에 쓰이는 固定 소수점 DSProcessor에는 적용시키지 못하는 것이다.

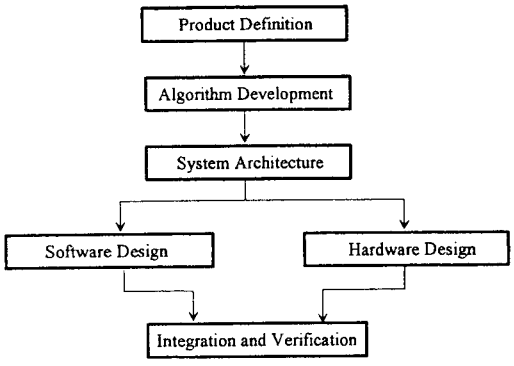
이런 家電기기 응용을 위한 固定小數點 DSP를 programming하는 일은, MCU의 경험을 갖고 있는 programmer에게도 아주 힘든 일이 된다. 이는 보통의 DSProcessor개발에 필수적인 實時間 시스템에 대한 개념과 algorithm에 대한 이해 이외, 固定 소수점 DSP가 갖고 있는 특수 용도 register들과 address mode, 제한된 memory크기에 맞는 programming을 위해 새로운 학습을 하여야 한다. 그리고, MCU개발을 위한 개발환경에 비해 不適切한 개발환경에 대해서도 직면하게 된다. 특히 고급언어를 이용한 programming은 固定小數點 DSP를 위한complier가 비효율적 이여서, 대부분의 program을 assembly에 의존하여야 한다. Assembly로 program하는 것을 돕기 위한 방법으로, 일부의 固定小數點 DSProcessor 에서는 C언어와 유사한 assembly언어 형태를 제공하기도 한다. 이런, 언어 tool의 불편 이외 실시간에서 debugging을 위해서는 emulation환경의 선행 개발이 필수적인 문제가 있다.

固定 小數點 DSProcessor를 위한 C complier의 효율성 증가를 위해서 특수 변수를 사용하여 처리하는 방법이 쓰이기도 한다. 이와 병행하여, digital신호처리를 위한 C언어의 보완도 병행하여 움직이고 있다. 이는 DSP를 위한 수식 연산 기능의 Library를 추가 시키는 방법과 multiplication-and-accumulation(MAC) operation, vector operation, string operation, memory management, hardware stack을 지원하는 C언어의 개량을 추진중이다.

V. DSP System개발 순서

〈그림 5〉에 DSP용 반도체를 개발하기 위한 순서를 나타내었다. 상품에 대한 기획단계에서 필요한 algorithm을 알아내고, 개발을 위한 설계방법을 정한다. 이때 필요한 algorithm의 개발은 일반적으로 고급언어를 사용한다. 초기에는 floating-point로 구현하여 algorithm의 이해와 여러 단위 algorithm간의 연결을 확인한다. 일반적으로 표준화된 algorithm의 경우에는 floating-point로 구현된 code가 標準의 일부로 公知된다. Algorithm의 개발은 公知된 것을 바탕으로, 응용에서 처리하려는 신호에 대한 분석으로부터 시작된다. 신호분석과 algorithm의 module화를 위해, 주요 연산 기능별 library(예 FFT, DCT, RLC 등)개발을 하게 된다. Algorithm개발의 최종 단계에서는 algorithm을 구성하는 각 변수의 길이를 有限하게 하여 algorithm에서 생기는 오차를 분석하게 된다. 이 단계를 integer simulation이라고 부른다. Algorithm개발단계에서 사용되는 고급언어는 보통은 C나 Fortran언어이다. 그러나, 이와 같은 언어는 control위주의 표현은 쉽게 구현되도록 되어 있으나, DSP에 필요한 수식 표현 및 memory할당 등의 표현에는 한계가 있다. 따라서, 최근에는 VHDL이나 Verilog HDL과 같은 hardware를 표현할 수 있는 언어와 고급언어를 같이 사용하는 방법을 취하기도 한다.

System architecture단계에서는 전체 system을



〈그림 5〉 DSP를 이용한 System의 개발 순서도

구현함에 있어, software와 hardware로 partition을 하고, hardware와 software간의 접속신호, 각 내부 블럭의 기능을 記述하는 설계를 진행한다. 이 단계에서는 설계를 위한 tool의 지원이 거의 없어, 대부분 사람에 의해 manually 진행된다. 이 단계를 거치면 hardware설계를 위한 behavior model과 software설계를 위한 flow chart 형태의 결과물이 나오게 된다.

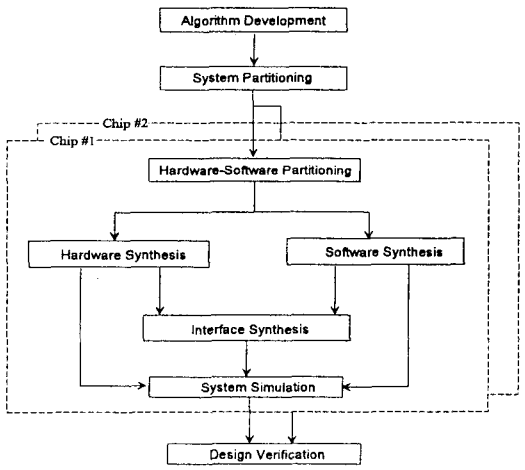
Software설계는 module library를 개발하거나 혹은 選定을 하고 구체적인 coding을 한다. 여기서, 고급언어의 지원이 가능한 DSProcessor의 경우는 compiler를 사용하게 되고 지원이 불가능한 경우는 assembly언어를 사용한다. Coding이 끝난 후에는 최적화작업 및 testing을 위한 단계를 거치게 된다. 최적화 작업은 사람에 의해 manually진행되는데, 노련한 경험이 절대적으로 요구되는 과정이다. Coding후 검증은 實時間 동작을 modelling한 환경에서 이루어 진다. Software설계와 병행하여 이루어지는 hardware설계는 보통의 LSI 설계와 동일한 방법이다. 다만, DSProcessor와 신호를 주고 받는 블럭에서는 software설계의 결과물과 동기를 해야 하기 때문에 software설계 부분과 상호 연관을 시켜야 한다.

Software와 hardware설계가 끝나게 되면 각각의 block을 묶어서 檢證을 실시하는데, 實時間 동작을 확인하기 위해 hardware로 구성된 emulation 방법을 주로 사용한다. Hardware emulation은 software 설계의 결과물은 DSProcessor의 In-Circuit Emulator를 통해 download되고, hardware는 FPGA로 구현되거나 특별히 제작된 emulation chip을 이용한다. 實時間 설계 檢證 단계에서 이루어지는 debugging환경은 boundary-scan emulation방법과 debugging환경을 實時間에서 동작시키는 위한 operating system(O/S)으로 이루어 진다. Boundary scan을 이용한 debugging은 근본적으로 어떤 상황이 일어난 것인지를 알아보기 위해서는 반드시 전체 DSP emulation system을 停止시킨 후에야 알 수 있는 근본적인 한계가 있다. 이를 극복하기 위해서는 real-time O/S가 background job으로 동작하면서 필요한 data를 모으고 host에 설치된 debugger와 통신하는 기능을 한다. 보통 이런 實時間 debugging환경은 다른 processor와의 호환성을 위해 각 DSP회사마다 동일한 protocol을 정의하여 사용한다.

VI. Co-Design

Embedded DSP반도체 개발을 위해서는 software와 hardware가 동시에 설계되어야 한다. 또한 각각의 설계가 통합된 환경에서 이루어져야 개발의 확실성 및 신속성이 보장되게 된다. 최근에 이를 위한 반도체 설계 Tool이 판매되고 있다. 대표적인 것으로, Mentor Graphics社의 DSP Station과 Synopsys社의 COSSAP, Cadence社의 SPW가 있다.

〈그림 6〉은 DSProcessor를 core로 사용한 반도체의 구조 설계 순서를 나타내고 있다. Algorithm의 개발이 끝난 후, 시스템 architecture단계에서 최초로 이루어 지는 것으로 전체 시스템을 몇개의 반도체로 구현할 것인가를 정하는 일이다. 여기서, 결정에 미치는 주요 요인은 전체 반도체의 가격인데 구체적인 요인으로 package pin숫자, 열방출 정도, testability, 유지 보수의 便宜性 등이 있다. 일단 시스템의 partition이 끝나고 나면, 各



〈그림 6〉 Co-Design 환경에서의 개발 순서

반도체를 hardware와 software로 partition하는 단계에 들어간다. Hardware설계에서는, 첫째로 각 부분을 analog로 할 것인지 digital로 구현 할 것인지를 결정, 둘째로 architecture에 대한 선택, 셋째로 각 register의 길이에 대한 결정, 넷째로 구현될 cell방식의 결정 등과 같은 단계로 구성이 된다. Software설계 또한, 어떤 core를 사용할 것인지, 固定小數點 표현을 어떻게 처리할 것인지, 각 software task의 역할 순서, memory와 peripheral I/O에 대한 결정 등의 단계를 거친다. 또한 hardware설계와 software설계 간에 접속부분의 설계가 있는데, 여기서는 주로 주고 받는 신호의 protocol을 결정한다. 이 모든 설계를 모아서 원하는 system규격을 만족하는지를 확인하는 절차를 거친다. 여기서, 잘못된 것은 각 설계 단계로 feedback되어 재설계된다.

이와 같은, hardware와 software를 동시에 설계하는 것을 Co-design이라 부르고 있다. 각각의 설계 단계에서 사용되는 언어가 다양해서 co-design을 지원하는 설계 tool에서는 여러 언어를 지원하도록 되어 있다. 일반적으로 사용되는 언어로는 hardware설계는 VHDL 혹은 Verilog HDL을, Software 설계는 C언어와 Assembly언어를 사용한다. 또한, software debugging을 위해서 해당 DSProcessor core에 해당되는 simulator도 동시에 제공되기도 하고, filter를 자동 합성하는 tool도 제공되기도 한다.

각 단계의 설계는 下位 階層으로 내려 갈수록 설계와 simulation에 소요되는 시간이 많이 필요하게 되어 가능한 上位 階層에서 완벽한 檢證을 하는 것이 중요하다. 이를 위해 각 설계 tool회사들은 system partition을 돕기 위해 functional level에서의 표준 library를 제공하고, 개념 설계와 계층적 설계를 지원하기 위해 graphic을 위한 editing 기능도 지원하여 반도체 설계의 경험이 없는 시스템 설계자가 접근이 쉽도록 하고 있다.

VII. 차별화와 低가격을 위한 ASIC DSP

현재의 家電機器에 점차로 multimedia기능의 추가됨에 따라 각 제품별의 차별성을 강조하는 점과 底가격이라는 두 가지 相反되는 요구가 존재하게 된다. 이를 충족시킬 수 있는 방법이 embedded DSP로 ASIC으로 구현하는 길이다. 이런 application-specific DSP-LSI(ASDSP)는 DSProcessor core에 특별한 기능의 hardware engine을 추가한 구성을 한다. 여기서, hardware engine은 응용에 필요한 DSProcessor의 MIPS를 최소화 할 수 있어, 향후 다른 기능을 추가에 대비하거나,DSProcessor를 낮은 주파수에 동작시켜 전력소모를 줄이는 역할을 한다.

이런 ASDSP를 위한 DSProcessor core는 작은 면적이 가장 중요한 선정 요소가 된다. 〈표 2〉는 현재까지 ASDSP에 적합한 DSProcessor의 일부를 소개하였다. 예로서, AT&T에서 개발한 DSP1618은 DSP1600의 core에 Viterbi decod-

〈표 2〉 ASIC을 위한 DSP Core의 例

Vendor	Core	Introduced Year	Data Width (bit)	Accumulator Width(bit)	Native Speed(MHz)	Core Area (mm^2)	Process (μ)
Clarspur Design	CD2400	1989	16	24	10	6	1.5
	CD2450	1994	16~24	32~48	50	3.9	0.8
DSP Group	Pine	1992	16	36	40	8.2	0.8
	Oak	1994	16	36	40	8.4	0.5
Sam sung	SSP1600	1990	16	24	20	3.6	0.8
	SSP1601	1992	16	32	25	3.8	0.8
SGS-Thomson	ST18932	1990	16	32	20	20	0.8
	ST18950	1994	16	40	40	10	0.5

ing과 bit처리를 할 수 있는 hardware engine을 추가시킴으로서 유럽방식의 digital cellular표준인 GSM을 구현하는데 사용되는 MIPS를 35MIPS에서 19MIPS로 줄일 수 있었다. 또 다른 ASDSP의 예로, Sega社의 16-bit game title인 Virtua Racing에서 사용한 경우가 있다. 이 ASDSP에서는 三星의 固定小數點 DSProcessor인 SSP1601에 game의 kernel에 해당되는 program과 graphic 加速器와 memory interface를 위한 hardware를 넣음으로서 game title의 無斷 복제를 원천적으로 막았을 뿐 아니라 3차원 game graphic을 실현하였다.

VIII. 맺음말

DSP기술은 미래를 새로운 기술임은 여러 분야에서 實證되고 있다. 또한, DSP기술을 개발에는 여러 분야의 기술이 종합적으로 필요하다는 면에서 종래의 다른 반도체 기술과는 相異하다. 따라서, DSP기술에 대해 전반적으로 설명하기는 불가능하지만 본 특집에서는 가능한 DSP기술의 발전을 全般的으로 記述하도록 노력하였다. 본 특집에 寄稿를 하도록 도움을 준 반도체 Micro본부의 모든 분들에게 감사를 드립니다.

참 고 문 헌

〔1〕 김재석, "고속 DSP기술동향," 전자공학회지, 제22권 제2호, pp.78~89, 1995년 2월

〔2〕 "Digital Signal Processing," Electronic Engineering Times, pp.43~79, July 17, 1995.

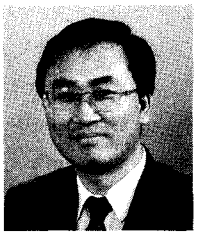
〔3〕 Will Strauss, *DSP Strategies for the '90s : The Multimedia Connection, Forward Concepts*, May 1992.

〔4〕 *Buyer's Guide to DSP Processors*, Berkeley Design Technology Inc., 1994.

〔5〕 *Design Tools & Methodologies for DSP Systems*, Forward Concepts, 1993.

〔6〕 Jaebum Hong and Il-Hyun Nam, 'Real-time implementation of time scale modification of speech signals with a 16-bit fixed-point DSP core," *Inter. Conf. on Signal Processing Applications & Technology*, Oct., 1995.

저 자 소 개



李 芳 遠

1959年 3月1日生
1979年 2月 서울대 전자공학과 학사
1981年 2月 KAIST 전기과 석사
1990年 5月 Univ. of Southern Calif. 전기과 박사

1979年 1月~1995年 현재 삼성전자 반도체 부문에 근무(理事 연구위원)
1992年 12月 IEEE Senior Member
1995年 7月 IEEE Trans. on Circuits & SystemII Associate Editor

주관심분야 : VLSI설계