

특별연재(6-5)

TMS320F2812와 IQ math 1편

- Floating Point Processors -

백 종 철

(주식회사 싱크웍스 이사)

흔히 접하는 DSP중에서 TMS320C3000 계열이나 TMS320C67계열은 Floating Point Processor다. Floating 이라는 말은 '등등 떠다니는' 정도로 번역될 수 있기에 Floating Point Processor를 부동(浮動) 소수점 연산자라 부르기도 한다. 한편, TMS320C2000 계열과 같은 DSP를 Fixed Point Processor라고 분류하는데, Fixed라는 말이 '고정된' 정도로 번역되기에 고정 소수점 연산자라 부르기도 한다.

도대체 뭐가 고정이고 부동이란 말인가? 참으로 알다가도 모를 일이다. 소수점이 등등 떠다녀서 부동이라는 말인가? 소수점이 어디에 콕 박혀, 빼도 박도 못해서 고정이라는 말인가? 참으로 부적절한 용어 선정이라 아니할 수 밖에 없다. 용어만 가지고도 충분히 이해할 수 있어야 할 텐데, 오히려 헷갈리게 만드니 어떻게 올바르게 할 수 있겠는가? 본 장에서는 Floating Point Processor와 Fixed Point Processor에 대한 설명과 아울러 적절한 용어를 제안하고자 한다.

1. Floating Point Processor

주위에서 DSP를 접해본 경험이 있으신 분들을 더러 뵈는 다. 어떤 칩을 사용해 보셨습니까? 하고 여쭙면, 대부분 TMS320C3X계열을 언급하신다. 거의 학교나 연구소에서 널리 쓰이는 듯 보인다. 왜, 3X를 사용하셨습니까? 하고 여쭙면, C3X는 Floating Point Processor이기에 수치 처리가 편리하고 연산이 정확해서 사용했습니다 하고 답을 하신다. 바로 Floating Point Processor의 가장 큰 장점이 되겠다.

왜 Floating Point Processor는,

- 수치 처리가 편리하고,
- 연산 결과가 정확하고,
- 다루기가 쉬운가?

이게 과연 정말일까? 남들이 얘기하니 그냥 그러려니 하고 무작정 사용하고 있는 것은 아닐까? 하나씩 따져보자. 지금도 3X를 사용하고 있다면, 상황은 조금 심각해질 수 있다. 3X는 TI에서 주력하는 3대 플랫폼에 포함되어 있지 않다. 왜 그럴까?

2. Floating Point Processor에서의 수치 표현

Floating Point Processor 내부에서는 숫자를 조금 특이하게 표현하고 있다. TMS320C67계열에서는 숫자를 어떻게 표현하고 있는지 그림 1을 통해서 살펴 보도록 하자. 참고로 C67XX 계열은 32비트 floating point 프로세서이며, 회로적으로 그림 1과 같이 데이터를 저장한다. 이를 흔히 단정도 (Single Precision) 표현이라고 한다. 솔직히 '단정도'라는 용어도 부적절하다는 생각을 지울 수가 없다.

좌측에서 우측으로 번호가 31부터 0으로 대입되어 있는데,

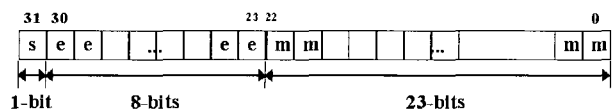


그림 1 TMS320C67XX 계열에서 데이터 표현법

각각은 모두 비트를 지시하고 있다. 각각에는 s,e,m 등의 문자가 보이는데 각각은 다음과 같다.

- s = 부호 비트 (sign bit)
- e = 지수 비트 (exponent)
- m = 밑 비트 (mantissa)

즉, 지수와 밑, 그리고 부호를 각각 표현하기에 소수점 표현이 가능하다. IEEE에서는 여기에 다음과 같은 추가적이 규칙을 정해놓았다.

- s = 부호 비트로 이 값이 0 이면 양수, 1이면 음수를 가르킨다.
- e = 지수로 -128 ~ 127 까지 표현한다.
- m = 23비트의 normalized fractional 형태로 숫자를 표현한다.

* normalized fractional이라는 형태는 분수형태로 수치를 표현한다는 뜻이다. 이어지는 수식을 보면서 이해를 도모해보자.

위와 같이 설명이 될 때, 그림 1이 표현하는 수는 어떻게 계산될까?

$$N = (-1)^s \cdot (1.m) \cdot 2^{(e-127)} \quad (\text{수식 1-1})$$

처럼 된다. m 에 대해서는 이어지는 설명을 기다려 주시기 바란다.

수식 1-1에서 e 값이 변화한다고 상상해보자. 소수점이 여기저기 돌아다니는 것을 쉽게 떠올릴 수 있을 것이다. 그래서, Floating Point Processor인 것이다. 필자는 이를 소수형 프로세서라 부르고 싶다. 칩 내부에서 소수 표현이 가능한 회로를 가지고 있기 때문이다. 연산자라는 말도 솔직히 싫다. 우리나라 말도 아니고, 어차피 외래어를 쓸려면 뜻 전달이 명확한 것이 낫다는 생각이다. 연산자가 뭔가? 참고로 Fixed Point 프로세서에는 소수를 표현할 회로가 없다. 다음 편에서 살펴 볼 예정이니, 조금만 기다려 주시기 바란다.

3. Floating Point Number의 규정

Floating Point Processor 끼리 데이터를 주고 받는 것을 가정해보자. 부호비트, 지수비트, 밑 비트들이 서로 맞지 않으면 데이터 교환에 낭패를 볼 수 있다. 그래서, IEEE에서는 754번째로 Floating Point Number에 대해서 규정을 해두었다. 그림 1이 바로 IEEE의 754 규격 중에서 32비트 데이터 표현 규격이다. 이를 단정밀도(Single Precision) 표현이라고도 한다고 했다. 따라서, TMS320C67계열은 IEEE의 754 규격을 따르고 있다. 그러나, C3X 계열은 TI 자체 규격

을 따르고 있기에 다른 프로세서와 Floating Number를 교환할 때에는 신경을 바짝 써야 한다.

3.1 단정밀도 숫자 표현

그림 1에서 표현 가능한 숫자 범위를 계산해보자. 아주 간단하다.

단정밀도에서 최대값

s = 0 ; 양수이어야 하니깐,

e = 255 ; 최대이어야 하니깐,

m = 1 - 2⁻⁽²³⁻¹⁾; 역시 최대이어야 하니깐

수식 1에다 대입해보자. 얼마가 나오는가?

$$\text{최대값} = [1 + (1-2^{-24})] \cdot 2^{255-127} = 3.4 \cdot 10^{38}$$

음의 최소값은 어떻게 될까?

각자 계산해보시 바란다. 답만 알려드리자면,

$$\text{최소값} = -3.4 \cdot 10^{38}$$

숫자 분해능도 계산해보시기 바란다. 최대값과 최소값의 범위를 한번 상상해보시기 바란다. 얼마나 큰가? 그래서, 좀처럼 소수형 프로세서에서는 숫자가 넘치는 현상이 발생하지 않는다. 많은 사람들이 좋아하는 이유 중 으뜸이다. 이를 'Dynamic Range가 넓다' 라고 표현한다. 당연히, Fixed Point Processor에 비해서 넓은 것이다. C 프로그램에서 float 는 데이터 형 대신에 double 이라는 데이터 형을 쓰는 이유를 잘 생각해보시기 바란다.

3.2 배정밀도 표현

C로 프로그램을 짤 때, 보다 높은 정밀도의 숫자가 필요해서 double이라는 키워드를 사용하는 경우가 있다. 두 배의 정밀도로 숫자를 표현하고 싶다는 것인데, IEEE 754에서는 배정밀도 수치 표현도 규정하고 있으며, C67XX 계열은 이를 따르고 있다. 그림 2는 배정밀도 규격이다.

두 배가 되었다고 해서, 지수가 16비트로 늘어난 것이 아님에 주의해야 한다.

3.3 IEEE 754 규격에서 추가적으로 알아 둘만한 내용

IEEE 754에서는 특별한 숫자까지 다루고 있다. 표 1을 우선 보도록 하자. 보면 아하~ 하게 될 것이다.

S(1)	E(11)	M(52)
------	-------	-------

그림 2 배정밀도(Double Precision) 표현

표 1 IEEE 754의 특별 숫자

s	e	m	Number
0	0	0	0
1	0	0	-0
s	0	≠0	$(-1)^s * 0.m * 2^{-126}$
s	0<e<255	m	$(-1)^s * 1.m * 2^{e-127}$
0	255	0	+∞
1	255	0	-∞
s	255	≠0	NaN (not a number)

아하~ 그렇구나! 맞는가? 프로그램 짜신 분들은 더러 겪어본 내용이 들어 있을 것이다. 32비트 단정밀도 표현일지라도 음의 무한대부터 양의 무한대까지 숫자 표현이 가능하다. Dynamic Range가 무한대인 셈이다. 참고적으로 32비트 Fixed Point Processor에서 Q31 형태로 수를 표현하면, Dynamic Range가 -1 ~ 1 에 불과하다. 대신에 정밀도가 매우 높기에, 제어 분야에 널리 쓰인다. 제어 분야는 Dynamic Range보다 정밀함을 요구하는 경향이 강하다.

4. 소수형 프로세서의 장단점

4.1 장점

쓰기가 너무나 쉽다. 누구말대로, 돌리면 되네~~ 다. 숫자에 대해서 고민할 필요가 없다. 그냥 쓰면 된다. 이유는? 소수형 프로세서의 Dynamic Range가 무한대이기 때문이다. 알고리즘 개발이 급선무인 곳에서 숫자처리 때문에 골머리를 싸맬다면, 이는 곧 낭비다.

4.2 단점


소수 표현을 위한 회로가 추가되기에 가격이 비싸다. TMS320C6211과 TMS320C6711은 서로 Pin for Pin 호환이 가능한 프로세서다. 다시말해 C6711 칩을 들어내고, C6211을 장착해도 잘 돌아갈 수 있다는 얘기다. 그런데,

C6211은 Fixed Point 프로세서이고 C6711은 소수형 프로세서다. 가격차가 50% 이상 난다.

어떻게 정수형 프로세서가 소수형 프로세서를 대신할 수 있을까? 소프트웨어적으로 처리한다. ANSI C 등에서 제공하는 math 함수가 이를 대신한다. 하지만 느리다. 그래서, 정수형 프로세서를 이용할 경우, 업체에서 제공해주는 수학 함수를 사용해야 한다. TMS320C/F28X 같은 경우에는 IQ math가 여기에 해당한다.

또한, 소비전력이 많다. 안 그렇겠는가? 회로가 더 들어갔는데... 결정적으로 느리다. 정수형 계산보다 소수형 계산이 더 많은 사이클을 요구한다. 소수형 프로세서에서 간단히 정수형 계산을 하는 경우를 볼 수 있는데, 이는 사이클 소비를 개선하기 위함이다.

한 가지 생각해볼만한 내용이 있다. Data Converter와 DSP와 연결문제인데, 데이터 컨버터에서 받아들이는 데이터는 한결같이 Fixed Point 데이터다. Floating Point Number를 받아들이는 Data Converter는 없다. 프로세서 성능만 고려할 것이 아니라, 시스템 전체 성능도 고려해야 한다.

소수형 프로세서에서 이러한 Data Converter로 값을 전달할 때 어떤 일들이 벌어지는지 각자 생각해보기 바란다. 비싼 값을 치르고 산 소수형 프로세서에서 실컷 계산을 하고서는 눈물을 머금고 비싼 사이클을 소비하면서 정수로 바꾸는 아이러니 벌어지는 현장을 목격하게 될지도 모른다. 

< 저 자 소 개 >



백종철(白宗哲)

1992년 KAIST 전기 및 전자공학과 졸업. 1994년 동 대학원 전기 및 전자공학과 졸업(석사). 1999년 현대오트넷(구 현대전자 전장사업본부) 근무. 2002년 Texas Instruments Korea 근무. 2002년 11월~현재 주식회사 싱크웍스 이사.