Executable Specification 기법을 이용한 MPEG Audio용 IMDCT 설계 및 기능검증

박 원 태(朴 洹 台), 조 원 경(趙 源 敬)
경희대학교 전자공학과
전화 : (0331) 201-3712 / 팩스 : (0331) 206-2942

Executable Specification based Design Methodology - MPEG Audio IMDCT Design and Functional Verification

Won-Tae Park, Won-Kyung Cho
Dept. of Electronic Engineering. Kyunghee Univ.
E-mail : warmpark@csvlsi.kyunghee.ac.kr

Abstract

Silicon semiconductor technology agree that the number of transistors on a chip will keep growing exponentially, and it is pushing technology toward the System-On-Chip. In SoC Design, Specification at system level is key of success. Executable Specification reduce verification time. This Paper describe the design of IMDCT for MPEG Audio Decoder employing system-level design methodology and Executable Specification Methodology in the VHDL simulator with FLI environment.

I. Introduction

반도체 공정기술의 발달로 면적당 집적도가 증가하여 하나의 칩에 시스템을 집적시키는 SoC(System-On-Chip)가 부각되고 있다. 이러한 system level의 chip Design에서는 기하급수적으로 복잡도가 증가하는데 반해서 더욱 단축된 개발기간을 요구하기 때문에 IP Based Design이라고 하는 새로운 개념의 설계방법이 강조되고 있다. 복잡한 알고리즘을 빠른 시간 내에 설계하기 위해서는 설계 전 단계에서의 Specification 이 중요하며, 이를 기반으로 하는 Spec-Based Design과 검증이 성공적인 설계의 요소가 되고 있다. 예전의 문서화된 Document Spec보다는 Test Bench에 바로 사용할 수 있도록 준비된 Executable Specification을 사용함으로써 verification 기간을 단축할 수 있다.

본 논문에서는 MPEG Audio Decoder의 중요 부분인 IMDCT 설계에서의 C 언어를 이용한 system-level Executable Specification 기술 방법과 FLI Environment을 이용한 verification을 통해서 효율적이고 복잡한 알고리즘을 빠른 시간 내에 설계할 수 있는 방법을 고찰해 보고자 한다.

II. Executable Specification

MPEG Audio 디코딩 알고리즘은 다양한 subblock들로 이루어지며, 이들이 순차적으로 수행된다. 앞 블록의 출력이 다음 블록의 입력으로 사용되므로 전후 블록들의 인터페이스를 고려해야 한다. 또한 fixed Point로 연산기를 설계할 경우 한정된 bit 때문에 오차가 발생하게 되므로 어느 정도의 성능(오차율)을 만족하는 연산기를 설계할 것인지, 발생한 오차가 다른 블록의 결과에 얼마나 영향을 미치는지 고려하여야 한다. 따라서 신뢰성 있는 IMDCT를 설계하려면 다른 subblock들과의 인터페이스, 오차 등을 고려해서 MPEG Audio Decoder의 Executable Specification을 기술한 후 IMDCT의 세부 구조에 대한 Specification을 기술한다. 본 논문에서는 System level에서 높은 추상화된

수준의 Specification이 가능한 C언어를 이용하여 기술하였다. 그림 1은 Executable Specification을 이용한 IMDCT의 Design Flow 보여준다.

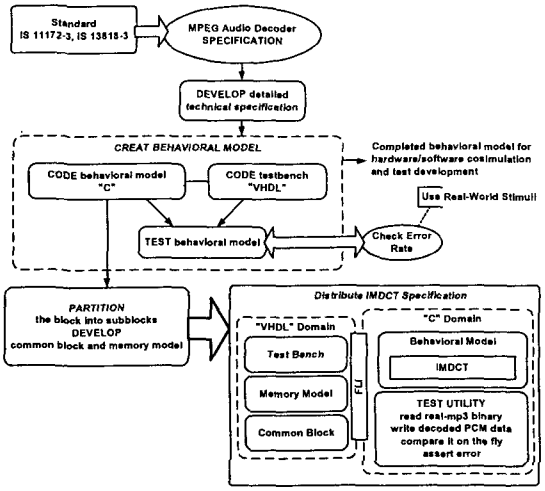

그림 1. Design Flow
Fig 1. Design Flow

MPEG 규정안을 기준으로 하여 MPEG Audio Decoder의 behavioral model(Executable Specification)을 C언어로 구현하였다. 이를 바탕으로 IMDCT를 포함한 subblock들을 분할하고, 분할된 subblock들을 floating-point와 fixed-point 두 가지 모델로 구현하여 error rate를 결정하였다. 각 subblock의 floating-point와 fixed-point 모델들을 shared object로 만들어서 한 모델에서 다른 모델을 사용할 수 있도록 기술하였다. 그림 2는 C언어를 이용한 각 subblock들의 specification을 보여준다.

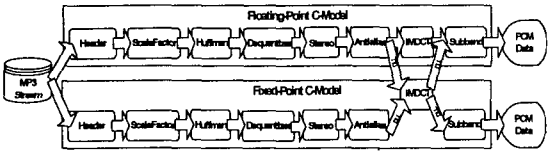

그림 2. C언어를 이용한 Specification
Fig 2. Specification using C Language

III. Executable Specification based IMDCT Design

기술된 Executable Specification을 기반으로 하여 IMDCT를 VHDL로 설계한다. Executable Specification에 의해 내부 연산기는 16bit로 설계하였으며, MPEG Audio Decoder 설계시에 다른 subblock에서 공유해서 사용할 수 있도록, MPEG Audio Decoder에 사용되는 모든 연산을 구현하였다. IMDCT는 크게 transform 처리 부분과 window 처리 부분으로 나눌 수 있는데, 두 부분의 연산 절차가 서로 상이하기 때문에 내부 제어기는 따로 설계하였고, master 제어기가 두 부분을 제어하도록 구현하였다. 그림 3은 설계된 IMDCT의 블록도이다.

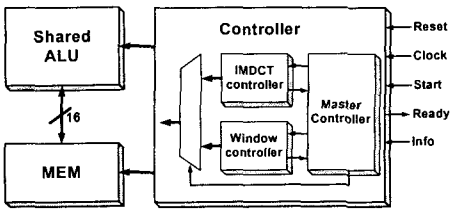

그림 3. IMDCT 블록도
Fig 3. IMDCT Block Diagram

III. Executable Specification을 이용한 Functional Verification

Executable Specification의 가장 큰 장점은 system level에서 고도의 추상화 수준을 갖는 specification을 Functional verification에 바로 사용할 수 있다는 점이다. 즉 system의 한 부분인 블록을 system level에서 verification 할 수 있다는 것이다. 본 논문에서는 Executable Specification을 testbench에서 바로 사용할 수 있도록 하기 위해서 FLI 환경에서 구현하였다. 그림 4는 FLI를 이용한 IMDCT verification 환경을 보여준다.

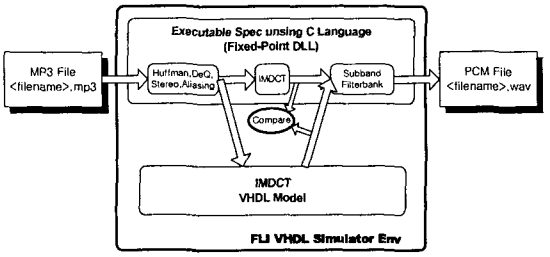

그림 4. FLI Test 환경
Fig 4. FLI Test Environment

FLI(Foreign Language Interface)는 순차수행언어인 C언어와 병렬처리를 기본으로 하는 VHDL simulator

사이에서 매개체 역할을 하여 C언어로 기술된 Excutable Spec과 VHDL model을 연동시켜 simulation하는 방법을 제공한다[5]. 그림5는 FLI 환경에서 VHDL모델과 C모델에서 서로 data를 주고받기 위한 Port mapping을 보여준다.

```
entity fli_IMDCT is                                    typedef struct {
port(
   filename   : in string(1 to 80);                    signalID  filename;

   Reset      : in STD_LOGIC;                          signalID  Reset;
   Clk        : in STD_LOGIC;                          signalID  Clk;

   Glbl_Gain  : out STD_LOGIC_VECTOR(7 downto 0);      driverID  Glbl_Gain;
   Preflag    : out STD_LOGIC;                         driverID  Preflag;
   Smpl_Freq  : out STD_LOGIC_VECTOR(1 downto 0);      driverID  Smpl_Freq;
   Sclfc_Scl  : out STD_LOGIC;                         driverID  Sclfc_Scl;
   Sclfc_Cmpr : out STD_LOGIC;                         driverID  Sclfc_Cmpr;
   Version    : out STD_LOGIC;                         driverID  Version;
   Stereo     : out STD_LOGIC;                         driverID  Stereo;
   Block_Type : out STD_LOGIC_VECTOR(1 downto 0);      driverID  Block_Type;
   Mode       : out STD_LOGIC_VECTOR(1 downto 0);      driverID  Mode;
   Mode_Ext   : out STD_LOGIC_VECTOR(1 downto 0);      driverID  Mode_Ext;

   IMDCT_Rd   : out STD_LOGIC;                         driverID  IMDCT_Rd;
   IMDCT_St   : in STD_LOGIC;                          signalID  IMDCT_St;

   D_Addr     : in STD_LOGIC_VECTOR(11 downto 0);      signalID  D_Addr;
   D_Bus      : in STD_LOGIC_VECTOR(15 downto 0);      signalID  D_Bus;
   D_ME       : in STD_LOGIC;                          signalID  D_ME;
   D_WEN      : in STD_LOGIC                           signalID  D_WEN;
);
end fli_IMDCT;                                         } inst_rec;
```

그림 5. VHDL과 C모델의 Port Mapping
Fig 5. Port Mapping between VHDL and C

Testbench에 그림 5의 오른쪽과 같은 entity를 가지는 블록을 선언하고 architecture에는 C모델의 경로를 기술한다. shared object로 구현된 C모델에서는 왼쪽과 같은 structure를 선언해주면 FLI 환경에서 서로 data를 주고받을 수 있다.

위와 같은 FLI 환경에서의 verification은 많은 이점들을 제공한다. 첫 번째 이점은 real-world stimuli를 이용할 수 있다. MPEG Audio Decoder 전체를 기술한 Executable Specification을 verification에 사용함으로써 IMDCT 이전 블록까지 복호된 MP3 데이터를 test vector로 사용하여 설계된 IMDCT를 simulation할 수 있고, 결과를 바로 Executable Specification의 결과와 비교하여 오류를 찾아낼 수 있다. 또한 VHDL simulation 결과를 Executable Specification에서 IMDCT 이후 block까지 처리하여 완전히 복호된 PCM data를 얻을 수 있으므로 설계된 IMDCT를 이용해서 복호된 음악을 귀로 청취할 수 있다.

두 번째 이점은 test vector를 생성하는 수고와 시간을 절약할 수 있다. 설계된 IMDCT는 MPEG I, II의 모든 sampling frequency(6가지), bitrate(14가지)를 지원한다. 따라서 전통적인 test 환경에서는 모든 경우를 테스트하기 위해서는 6×14=84가지의 test vector를 text file로 만들어야한다. 여기에 스테레오 모드가 4가지 이상 존재하므로 300개가 넘는 test vector를 만들어야 하는 번거로움이 있다. 또한 부호화된 MP3 data가 프레임으로 구분되어 있고, 한 프레임이 2개의 granule로 나뉘어 진다. 1초 정도의 음악에 40개 정도의 프레임이 존재하기 때문에 80개(granule 단위로)의 test vector를 만들어야 하므로 전체적으로 300×80=24000개가 넘는 test vector 파일을 만들어야 하므로 많은 노력과 시간이 소모된다. 하지만 FLI 환경에서 C언어로 기술한 Executable Specification을 이용하는 verification 방법에서는 부호화된 MP3 data를 binary로 읽어들여 graule 단위로 test vector를 넘겨줄 수 있기 때문에 이러한 수고를 덜어주고 시간도 절약할 수 있다.

세 번째 이점으로 MPEG Audio Decoder 설계시 verification에 소모되는 시간을 절약할 수 있다. 설계 초기단계에서 다른 subblock들과의 인터페이스 등을 고려하여 기술한 시스템수준의 Executable specification을 verification에 그대로 사용함으로써 system level에서의 verification을 할 수 있다. 따라서 MPEG Audio Decoder 설계에서는 설계된 IMDCT를 다른 subblock들과의 integration 작업만 수행하면 된다. Executable Specification을 이용한 설계방법의 또 다른 장점은 subblock의 설계를 동시에 수행할 수 있다는 것이다. MPEG Audio Decoder의 각 Subblock들이 순차적으로 수행되지만, 모든 subblock들을 system level에서 기술한 Executable Specification에 의해 verification 할 수 있으므로 이전 block의 설계에 관계없이 다음 블록의 설계도 동시에 할 수 있다. 그림 6은 MPEG Audio Decoder 설계시에 verification에 사용할 수 있는 통합 simulation 환경을 보여준다.

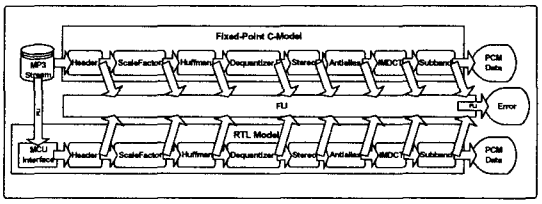

그림 6. MPEG Audio Decoder 통합 simulation 환경
Fig 6. Integrated Simulation Environment of MPEG Audio Decoder

모든 subblock의 C 모델의 결과와 VHDL 모델의 simulation 결과를 FLI 환경에서 바로 비교할 수 있으므로 전체 system integration에서 어느 블록에서 오류가 발생하는지 쉽게 찾을 수 있고 verification에 소모되는 시간도 줄일 수 있다.

IV. 결론

반도체 공정기술의 발달은 기하급수적인 집적도의 증가를 가져오고, 이는 한 칩에 System을 모두 집적시키는 SoC설계로 설계방법을 변화시키고 있다. SoC 설계에서는 system level에서의 Specification이 성공의 중요한 요소가 되고 있다. 본 논문에서는 MPEG Audio Decoder에서 사용되는 IMDCT를 system-level Executable Specification에 의해 설계해 보았다. C언어를 이용하여 MPEG Audio Decoder의 Executable Specification을 기술하였고 FLI을 이용하여 IMDCT의 Functional verification에 Executable Specification을 그대로 이용하였다. 이 방법은 real-world stimuli를 이용할 수 있고, test vector 파일을 생성하는데 소모되는 시간과 수고를 덜어주며, MPEG Audio Decoder 설계시에 verification에 소비되는 시간을 단축시켜준다. 모든 블록들이 시스템 수준으로 기술된 Executable Specification을 이용하여 설계되고 verification되었다면 최종 system 설계에서는 각 블록들을 신뢰할 수 있으므로 integration 작업만 하면 된다. integration이 끝나고, 전체 시스템의 functional verification에도 Executable Specification을 사용할 수 있으므로 빠른 시간 내에 신뢰성 있는 설계를 가능하게 해 준다.

참고문헌

[1] ISO/IEC JTC1/SC29/WG11 No.71, Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s Part 3 : Audio(ISO/IEC 11172-3) , Mar. 1993.

[2] ISO/IEC JTC1/SC29/WG11 No.803, Generi c Coding of Moving Pictures and Associated Audio : Audio(ISO/IEC 13818-3) , Nov. 1994.

[3] D. Gajski, F. Vahid, S. Narayan, J. Gong, *Specification and Design of Embeded Systems,* New Jersey, Prentice Hall, 1994.

[4] A. Gerstlauer, S. Zhao, D. Gajski, A. Horak, *Design of a GSM Vocoder using SpecC Methodology,* SpecC Technical report.

[5] ModelSim User's Manual, ModelTech