

DVD Servo용 IC개발에 적용한 TeakLite core 기반의 Seamless CVE 환경

서승범, 안영준, 배점한
삼성전자 중앙연구소 NS Lab.
전화 :031-200-4770/ 핸드폰 :011-711-8705

Seamless CVE Environment Using TeakLite Core for DVD Servo

Seung-Bum Suh, Young-Jun Ahn, Jum Han Bae
NS Lab., Corporate R&D Center, Samsung Electronics
E-mail : sbsuh@samsung.com

Abstract

Verification is one of the most critical and time-consuming tasks in today's design process. This paper describes the basic idea of Co-verification and the environment setup for the design of DVD Servo with TeakLite DSP core by using Seamless CVE, Hardware/Software Co-verification tool.

I. 서론

오늘날 디자인의 복잡성과 개발 기간의 증대는 embedded system 설계를 위한 새로운 방법론을 필요로 하게 되었다. 기존의 embedded system 설계는 독립된 하드웨어, 소프트웨어 틀을 이용하여 개발이 이루어졌다. 그러나 이와 같은 방법은 소프트웨어 설계와 검증이 하드웨어 prototype이 완성된 후에 이루어질 수 밖에 없다는 시간적, 물리적 제약성을 지니고 있다. 이를 극복하기 위하여, 하드웨어, 소프트웨어 설계가 동시에 이루어질 수 있도록 하는 Co-Verification이 제기되었다. Co-Verification

의 핵심은 소프트웨어 시뮬레이션이 충분한 성능을 발휘할 수 있도록 하는데 있으며, 이를 위해 Seamless CVE는 다양한 Seamless CVE의 하드웨어 모델과 최적화 방법을 도입하고 있다.

본 논문은 Co-Verification 틀인 Seamless CVE를 사용할 수 있는 개발 환경을 제공함으로써 실제 IC 설계 및 검증의 토대를 마련하고자 한다. 구체적인 환경은 현 개발 중인 DVD Servo용 IC 개발에 Co-Verification tool인 Seamless CVE를 적용할 수 있도록 하고 있다. DVD Servo 개발에 TeakLite DSP core를 이용하며, 하드웨어 시뮬레이터로는 VCS를 소프트웨어 시뮬레이터로는 DSP Debugger를 이용한다.

II. Co-verification

기존의 embedded system의 디자인은 High-level system design, Detailed design and implementation, Hardware/Software integration의 과정을 거친다.

기존의 설계 방법에서 hardware/software integration은 hardware implementation이 완료된 후에 이루어진다. 이에 반해 Co-verification은 하드웨어

어 시뮬레이션과 소프트웨어 시뮬레이션의 링크를 통하여 가상 프로세서에서 소프트웨어를 실행할 수 있도록 한다. 따라서 hardware/software integration을 위해 hardware prototype을 기다릴 필요가 없게 된다.

Seamless CVE는 Co-verification의 성능 향상을 위하여 프로세서의 function과 interface를 분리하며, 하드웨어 시뮬레이션 시 버스 사이클을 선택적으로 제거하도록 한다

Seamless CVE의 PSP(Processor Support Package) 및 Seamless CVE Memory Model 등은 시뮬레이션의 효율을 증대 시키기 위하여 이용된다. 또한 Data access optimization, Instruction fetch optimization, Time optimization의 다양한 optimization 방법을 제공하여, 시뮬레이션 속도를 향상시킬 수 있다.

III. Co-verification 환경 설정 - Seamless CVE

Seamless CVE를 이용하여 co-verification을 하기 위하여 다음과 같은 사항이 요구된다.

1) Seamless CVE installed

CVE_HOME 환경 변수가 Seamless CVE installation tree의 top을 가리킬 수 있도록 한다.

2) Licensing

Seamless CVE와 함께 제공된 모델들을 이용하기 위하여, MGC_HOME 변수는 최신의 Seamless CVE 라이선스를 지닌 MGC home tree를 가리키도록 한다. 또는 MGLS_HOME 변수가 \$CVE_HOME/mgls를 가리키도록 한다.

3) Seamless CVE cosimulation models in your design

Cosimulation을 이용하기 위하여는 디자인에 이용된 Seamless CVE processor model이 삽입되어야 한다. Seamless CVE Processor Support Packages(PSPs)가 이 모델을 가지고 있으며, 디자인에 이용된 각각의 메모리들에 data-access

optimization을 적용하여 cosimulation 하기 위해서는 Seamless CVE memory model들이 이용되어야 한다.

4) Hardware simulator

Seamless CVE cosimulation과 compatible한 하드웨어 시뮬레이터가 필요하다.

5) Instruction Set Simulator (ISS) 또는 Host Code Execution (HCE) program

XRAY Simulator/Debugger가 기본 Seamless CVE를 위한 Instruction Set Simulator(ISS) 이다. 몇몇 프로세서들은 별개의 ISS가 이용 가능하다. 특정 프로세서를 위한 각각의 ISS setup 정보는 각각의 프로세서에 대한 the Seamless CVE PSP Data Manual에서 얻을 수 있다.

XRAY Simulator/Debugger를 사용하기에 앞서 다음과 같은 환경 변수를 설정해 주어야 한다.

1) DISPLAY는 workstation display를 가리켜야 한다.

2) LM_LICENSE_FILE은 XRAY software를 실행하기 위한 라이선스를 가리켜야 한다..

3) XRAY는 compiled object 또는 ISS를 시뮬레이션 되어 할 source code를 가리켜야 한다.

4) LD_LIBRARY_PATH 또는 SHLIB_PATH Seamless CVE installation tree의 dynamic shared library들을 가리켜야 한다.

3.2 Hardware Simulation Setup - VCS

Seamless CVE 와 Chronologic VCS (VCS)의 환경 설정은 다음과 같은 과정으로 이루어진다.

1. Seamless CVE cosimulation module의 Verilog version을 instantiate한다.
2. 필요시 <psp>_compile_vcs.sh file을 수정한다.
3. <psp>_compile_vcs.sh script를 실행한다.
4. Seamless CVE를 실행하여 VCS를 불러드린다.

표 1. Seamless CVE 환경변수

CVE_HOME	The top of the Seamless CVE installation tree
MGLS_HOME	\$(CVE_HOME)/mgl (license server의 위치)
MGLS_LICENSE_FILE	License file의 위치

시작에 앞서 표1과 같이 환경 변수들을 설정해 주어야 한다.

현재 Seamless CVE와 VCS 사용을 위한 환경 설정은 아래와 같다.

표 2. Seamless CVE 환경변수 설정

```
#setenv CVE_HOME
/home1/EDA/seamless3/cve_home.ss5
setenv CVE_HOME /home1/EDA/teakl/cve_home.ss5
setenv MGLS_HOME
/home1/EDA/seamless3.1/cve_home.ss5/mgl
setenv MGLS_LICENSE_FILE 1717@asics24
set path = ($CVE_HOME/bin $MGLS_HOME/bin $path)
```

Cosimulation을 위해 디자인에 이용된 Seamless CVE processor model을 우선 instantiate해야 한다. 뿐만 아니라 Seamless CVE의 performance optimization을 위해서는 관련된 모든 메모리에 대해 Seamless CVE memory model을 instantiate해야 한다. 각각의 모델은 port interface로 정의된 Verilog source file과 Seamless CVE로의 PLI link인 shared library로 이루어져 있다.

CVE cosimulation model을 위한 Verilog source file들은 \$CVE_HOME/vlog에 위치한다. 이 directory는 processor model (예, teakl.v)와 generic memory model (예 sram.v)의 source file을 지니고 있다. Shared library들은 \$CVE_HOME/lib에 위치한다.

표 3. TeakLite Seamless CVE model을 위한 PLI link

Seamless CVE	\$(CVE_HOME)/lib/libcve_vlog.so
--------------	---------------------------------

Seamless CVE Memory	\$(CVE_HOME)/lib/libcve_mem_vlog.so
PSP for the CPU	\$(CVE_HOME)/lib/teakl_vlog.so

모든 Seamless CVE model들은 user-configurable attribute들을 지니고 있다. 예를 들면, Seamless CVE memory model들은 address와 data-bus의 width를 NUM_ADDR_BITS와 NUM_DATA_BITS parameter들의 설정을 통하여 정의할 수 있도록 한다. 각 모델들의 Verilog source code들은 각 parameter를 위한 default 값들을 지니고 있으며, *defparam* statement를 이용하여 새로이 정의할 수 있다.

VCS를 Seamless CVE와 이용하기 위해서는 Seamless CVE를 위한 Verilog PLI 함수들을 포함한 VCS executable file을 build해야 한다. *<psp>_compile_vcs.sh* script를 이용하여 VCS executable을 build한다. The *<psp>_compile_vcs.sh* script는 \$CVE_HOME/vcs에 위치하며 \$CVE_HOME/vcs에 위치한 *<psp>_pli.tab* file들을 이용한다. 각각의 *<psp>_pli.tab* file은 Bus Interface Model, generic memory model, the Seamless CVE cosimulation kernel VCS executable binding하는 역할을 한다.

VCS와 Seamless CVE를 위한 *<psp>_compile_vcs.sh* file은 다음과 같다. (TeakLite)

표 4. VCS 실행 파일 생성 위한 Script (TeakLite)

```
$VCS_HOME/bin/vcs $* \
-P $CVE_HOME/vcs/teakl_pli.tab \
$CVE_HOME/vlog/teakl.v \
$CVE_LD_FLAGS -L$CVE_HOME/lib -lteakl_vlog
-lcve_mem_vlog -lcve_vlog
```

<psp>_compile_vcs.sh script를 실행하여 *simv*라는 실행 file을 생성한다.

VCS executable file 생성 후, Seamless CVE를

실행한다. Cosimulation 설정 시 하드웨어 시뮬레이터를 불러드려야 한다. VCS invocation 설정 시, VCS executable file (default: simv) 명시해 준다.

3.3 Configure the Processor

Cosimulation을 위한 Seamless CVE processor instance의 설정은 다음과 같은 과정으로 이루어진다.

1) Set up the software simulator invocation

Seamless CVE Session window의 Software Simulation icon을 클릭하여 실행한다. TeakLite PSP는 OakDSPCore Debugger를 이용하며 Host Code Execution(HCE)을 지원하지 않는다.

2) Set up memory instance mapping

Map Memory Instances icon을 클릭 하여 Memory Map dialog box를 실행한다. Base address와 bit position의 기본설정과 필요시 Buswidth override와 interleave를 설정한 후 Memory Map dialog box를 닫는다.

3) Set up the memory-access ranges

Memory Access Ranges icon을 클릭한다. Memory Ranges dialog box가 나타나면, 하드웨어 디자인에 맞추어 각각의 설정 값을 입력해준다. 메모리의 처음과 마지막 주소를 수정한다. non-optimizable 메모리 영역은 Software-only, Hardware-only 또는 Illegal의 영역으로 설정하여 준다. 이와 같은 영역은 Seamless optimizable memory instance가 mapping된 영역에는 이용될 수 없다. Initial read/write cycle과 burst cycle을 위한 wait state를 지정해준다.

한계에 이르렀으며, 이에 대한 대안으로 Co-verification이 등장하였다. 이러한 시점에서 현재 개발중인 TeakLite를 이용한 DVD Servo 개발에 Co-verification tool인 Seamless CVE를 이용하고자 했으며, 본 논문에서는 이를 위한 환경 설정 방법을 제시하였다. 현 시점에서 Co-verification tool의 효율성 제고를 위한 환경 설정 만을 제시하고 있으나, 이를 바탕으로 DVD Servo을 개발을 마칠 즈음에는 실질적인 Co-verification의 유효성을 검증할 수 있을 것으로 기대된다.

참고문헌

- [1] Mihai Manolescu, Igor Furlan, "Software/Hardware cosimulation methodology", 1999 IEEE
- [2] Thomas W. Albrecht, Johann Notbauer, Stefan Rohringer, "HW/SW CoVerification Performance Estimation & Benchmark for a 24 Embedded RISC Core Design", 1998 ACM
- [3] Johann Notbauer, Thosmas Albrecht, Georg Niedrist, Stefan Rohringer, "Verification and Management of a multimillion-gate embedded core design", 1999 ACM
- [4] Arnold S. Berger, "Processor Modeling And Software Execution in HW/SW Co-development Environments", Applied Microsystems Corporation
- [5] Serge Leef, "A Methodology for Virtual Hardware /Software Integration", Mentor Graphics Corporation
- [6] Russ Klein, "Hardware/Software Co-Simulation", Mentor Graphics Corporation

IV. 결론

오늘날 복잡한 embedded system의 개발에 있어서, 기존의 FPGA prototype 검증 방법은