

고속 아키텍처 탐색을 위한 transaction-level bus functional model 기법

김 시현, 설 선미, 정 의영, 최 규명, 공 정택, 어 수관
DS 총괄 SYSTEM LSI 사업부 SOC 연구소 CAE 센터
경기도 용인시 기흥읍 농서리 산 24 번지삼성전자(주)
+ 82-31-209-4255

Email: {si-hyun.kim , sm.seol, euiyoung.chung, kmchoi, jkong, sookwan.eo}@samsung.com

Abstract - IP reuse is a crucial methodology for large-scale SoC designs and their performance critically depends on the quality of the communication architecture. Due to the large search space of the communication architecture, its optimal design requires a new methodology to measure the performance of each architecture instance very quickly with reasonable accuracy. This cannot be achieved with the traditional slow RTL simulation. To tackle this issue, we propose a novel transaction-level BFM (Bus Functional Model) method which only considers the communication behavior for the fast simulation. We classify IPs into four classes and provides a BFM creation method in an automated way for each class. Our method considers the inter-play effect between IPs and, the real-time constraint behavior of a IP can be easily modeled. To enhance the accuracy, we introduce an operation-mode based characterization method which incurs multiple BFMs for a single IP. The entire system is created by selecting BFMs according to the given top-level operation scenario (typically, worst-case scenario) and then connecting them to the RTL communication network. The proposed method was applied to 3 million gate one-chip DVD player and the results show that the full-chip simulation with our BFMs only took less than 1 hour and the accuracy of analysis is within 90% compared to RTL simulation.

Keywords: Communication architecture, BFM, transaction, performance analysis.

I. 서론

대규모 회로의 효율적 설계를 위한 방안으로 재사용 가능한 IP를 이용한 플랫폼 based 설계 방법이 대두되고 있다([1],[2]). 이 기법은 계산 유닛의 재사용을 통해 설계 생산성을 높일 수 있는 반면, 통신 아키텍처의 효율적 설계라는 또 다른 과제를 가지고 있다. 따라서, bus layer의 변화에 의한 bus utilization, throughput, latency 등을 효율적으로 분석할 수 있는 방법이 필요하다([3],[4]). 기존의 RTL based 플랫폼은 성능평가 결과를 실제 실장과 비교하여 비교적 작은 오차 내에서 분석할 수 있는

반면, 저속의 시뮬레이션 속도로 인해 다양한 통신 아키텍처에 대한 성능분석은 현실적으로 어려운 실정이다([5]).

본 논문에서는, RTL IP의 동작과 비교하여 최대한 높은 정확도를 가지면서 고속 시뮬레이션이 가능한 새로운 시뮬레이션 모델링 방법론을 제안하였다. 간략히 제안된 방법론을 설명하면, 각 master 블록에 대한 transaction 단계의 RTL 시뮬레이션으로부터 동작정보를 추출하고, 추출하기 어려운 경우에는 설계자로부터 직접 받아들이며, RTL IP의 통신관련 동작만을 묘사하는 Bus Function Model (BFM)을 이용하는 것이다. 하나의 master 블록은 여러 동작모드에 대해 각각의 BFM을 생성하고, top-level 플랫폼은 설계자가 구상한 시나리오에 적합한 BFM들을 선택하여 구축된다. 구축된 top-level 플랫폼의 성능분석을 효율적으로 하기 위해서 bus latency, bus contention, 그리고 throughput을 분석하는 환경도 개발되었다. 본론에서는, BFM의 개발 방법, 모델링 특성, 성능분석 방법을 설명하였고, 실험 결과에서는 실제 BFM을 이용한 top-level 최악조건 시뮬레이션을 통한 결과를 바탕으로 BFM의 성능과 정확도를 분석하였다.

II. 본론

1. Bus Functional Model (BFM) 개요

플랫폼상에서, bus utilization과 각 IP별 bandwidth 분석은 상당히 중요한 성능평가 사항이며, 가능한 많은 시나리오를 쉽게 구성하고 적용하여 고속으로 평가하기 위해서는 각 RTL IP를 고속의 bus transaction 레벨에서 모델링 하는 방안이 필요하다.

RTL IP의 동작은 그림 1과 같이 communication (bus interface)부분과 computation (data process) 부분으로 구성된다. BFM은 RTL bus master IP의 communication 부분, 즉 bus transaction 동작을 IP의 동작 모드 별로 모델링 하여 bus 레벨에서 성능 평가를 하기 위한 시뮬레이션 모델이다.

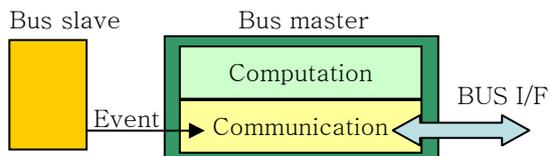


그림 1. Bus master의 기능적 분리 방법

Bus transaction은 그림 2와 같이 초기 idle 구간, n burst 만큼의 burst transaction 구간에서 각 transfer protocol, 그리고 burst transaction간의 control 구간에서 동작 특성을 정의할 수 있다. 각각의 구간에서 동작을 결정하는 신호들을 정확히 분석한 후에 모델의 parameter set을 결정함으로써 모델의 정확도를 높일 수가 있다.

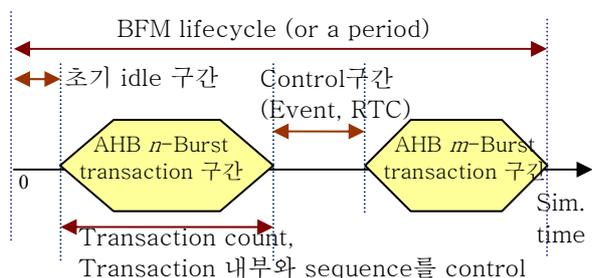


그림 2. Bus transaction 동작 특성

그림 3은 BFM block diagram으로서, 초기 idle구간, Real-Time Constraint(RTC), event를 control 하기 위한 신호들이 AHB port외에 입력신호로 추가 되었다. Output 신호인 HADDR, HBUSREQ, HSIZE, HBURST, HWRITE 신호는 각각 address, bus request, data size, burst type, 그리고 data transfer 방향을 의미한다. BFM은 bus 신호를 생성하는 부분과 transaction 레벨로 bus communication을 수행하는 transaction adaptor로 구성된다.

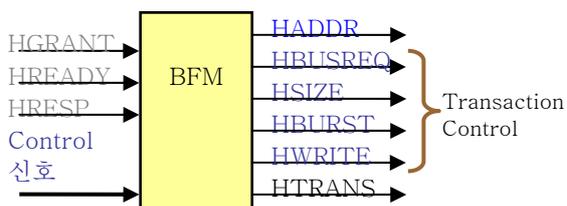


그림 3. BFM block diagram

2. BFM 개발 Flow

그림 4의 BFM 개발 flow는 RTL 시뮬레이션에서 모니터를 이용하여 bus transaction protocol과 sequence를 추출한 결과와 control 신호의 property 정의와 모델링 방법론에 관하여 설계자가 지정한 design configuration을 동시에 고려한 모델링 방법을 적용하였다.

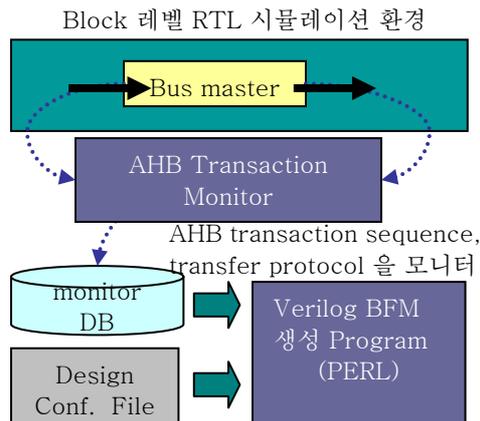


그림 4. BFM 개발 Flow

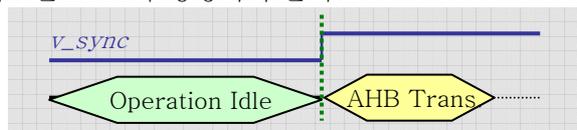
그리고 모니터에서 추출된 transaction 동작을 이용한 모델링에는 각 BFM의 output 신호별로 그 특성에 맞게 4개의 모델링 방법 중 하나를 선택하여 동작을 모델링 할 수 있어 RTL communication 동작과 비교하여 높은 정확도를 가지는 BFM 개발 방법을 적용하였다.

A. BFM input control신호의 모델링 방법

BFM control의 property는, 초기 idle, event, RTC 구간으로 분류하고, 각 3개의 property 의미와 design configuration에서의 기술 방법을 설명한다.

초기 idle구간은, BFM이 처음으로 bus transaction (bus access)을 시작하기 위한 조건을 의미한다. 조건은 configurable register가 setting되고 computation cycle이 지난 후에 시작되는 경우와 다른 IP에서부터의 신호가 활성화된 후 시작되는 경우의 두 가지로 나눌 수가 있다.

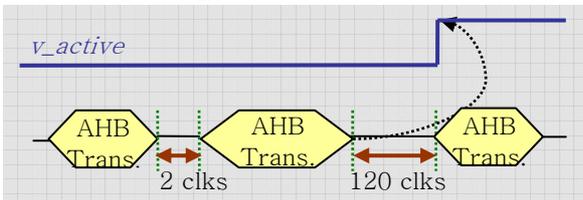
전자의 경우는, BFM에 *bfm_enable_x*와 같은 가상신호를 만들어, 테스트 벤치상에서 다른 BFM과의 시작 sequence를 고려한 후, 가상신호를 활성화 시키는 방법으로 초기 idle구간을 control할 수가 있다. 후자의 경우는, 그림 5와 같이 bus slave에서 드라이브되는 control 신호 *v_sync*가 active된 후 bus access가 발생하는 경우를 *CONT_PORT*와 *INIT_WAIT* 조건을 design configuration file에서 기술하면, BFM 생성 프로그램은 bus slave에서의 *v_sync* 신호를 모니터하여 처음 bus access를 생성하는 control 논리 회로를 BFM에 생성시켜 준다.



```
CONT_PORTS = " v_sync(1) "
INIT_WAIT   = " v_sync == 1' b1 "
```

그림 5. BFM control 방식: *INIT_WAIT*

시뮬레이션 모니터 결과에서 추출되는 bus transaction간의 interval은 bus traffic 상황에 따라 변할 수 있으며, 이러한 경우의 interval은 어떠한 외부 event를 기다리고 있는 경우로 mapping할 수가 있다. 예를 들어, 그림 6에서 실제 transaction간의 interval은 120 클록 이지만, pre-transaction이 칩 레벨 시뮬레이션에서의 bus traffic 상황에 따라 그 값이 변할 수 있다. 즉, 120 클록은 실제로 *v_active* 신호의 event를 기다리기 위한 interval이며, 이러한 경우, 100 클록 이상의 interval은 event를 기다리는 조건으로 표현하면 정확한 동작의 모델링이 가능하다. Design configuration file에서 *CONT_PORT*, *EVENT_CYCLE*, *EVENT_MAP* 조건을 기술 하면, BFM 생성 프로그램은 모니터에서 추출된 120 클록 대신에 *v_active* 신호를 모니터 하여 다음 bus transaction을 생성하는 control 논리 회로를 BFM에 생성시켜준다.

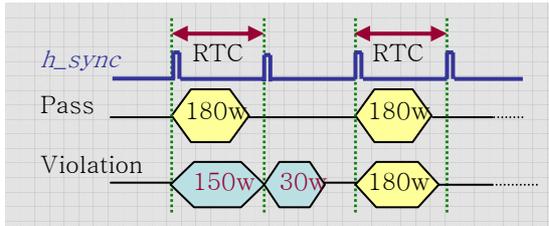


```
CONT_PORTS = " v_active(1) "
EVENT_CYCLE = " 100 "
EVENT MAP = " v_active == 1' b1 "
```

그림 6. BFM control 방식: *EVENT_MAP*

RTC의 동작은, 정해진 시간 내에 필요한 양의 data를 전송한 경우와 hard bus traffic에 의하여 전송하지 못한 경우로 구분하여 burst transaction간의 interval을 control 해야 한다.

예를 들어, 그림 7과 같이, *h_sync* 구간에 180 word 만큼의 data를 전송하는 것이 RTC라고 가정하자. 정해진 시간 내에 전송이 완료 되면 다음 RTC 조건이 발생할 때까지 전송을 기다리고, 전송을 완료하지 못한 경우에는 남은 data 양을 interval 없이 전송하도록 *CONT_PORT*, *RTC_CASE*의 조건을 design configuration file에서 기술 하면, BFM 생성 프로그램은 RTC 구간내의 실제 전송된 data 양을 카운트하면서, burst transaction을 control 하는 논리 회로를 BFM에 생성시켜준다.



```
CONT_PORTS = " h_sync(1) "
RTC_CASE = " 180, h_sync, 1 "
```

그림 7. BFM control 방식: *RTC*

B. BFM output 신호의 모델링 방법

BFM output 신호는 기본적으로 AHB bus traction의 모니터에서 추출된 transaction간의 interval (*HBUSREQ*를 모델링)과 transaction control 신호(*HBURST*, *HSIZE*, *HWRITE*를 모델링)로 모델링 할 수 있다. 하지만, 모니터 결과의 값들을 단순히 trace 하여 모델링을 하면 실제 IP의 동작과 다른 경우가 발생할 수도 있으며 다양한 테스트 시나리오를 구성하기 위해서는 쉽게 그 동작을 조절할 수 있어야 한다. 그래서 BFM output 신호의 모델링 방법을 deterministic, non-deterministic, trace-driven 그리고 user-defined method의 4개의 방법으로 구분하여, 각 IP의 특성에 맞게 모델링 할 수 있도록 하였다.

Deterministic Method(DM)는 transaction 간의 interval을 모니터에서 추출하여 bus request interval *HBUSREQ* 신호의 동작을 FSM 스타일로 모델링 하는 방법이다. FSM은 그림 8과 같이 bus transaction을 발생하는 기능을 state로, transaction interval과 transfer count를 state transition 조건으로 mapping하여 구성된다. DM의 특징은 시뮬레이션 결과를 정확히 모델링 할 수 있다는 것이다. BFM 생성 프로그램은 *INIT_WAIT*, *EVENT_MAP*, *RTC* 등의 transition control 방식을 고려하여 state transaction이 수행되도록 BFM에 control 논리 회로를 생성하므로 실제 RTL의 bus request interval의 정확한 모델링이 가능하며, transfer count와 transaction interval을 parameter 형식으로 쉽게 수정할 수 있는 Verilog format으로 BFM이 생성됨으로 BFM의 flexibility을 높일 수 있는 방법이다.

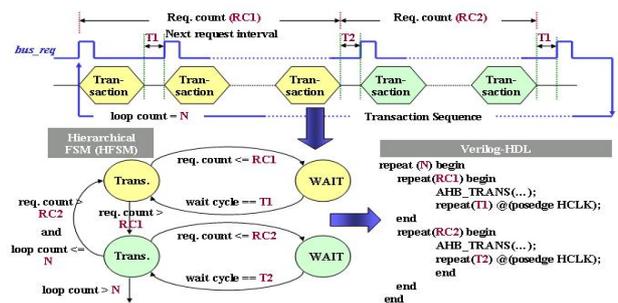
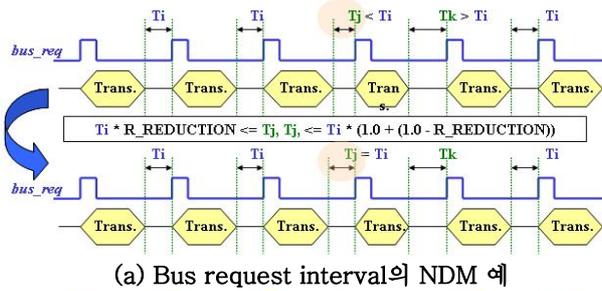


그림 8. BFM output 모델링 방식: *Deterministic*

Non-Deterministic Method(NDM)는 BFM output 신호의 동작을 평균적이고 확률적으로 모델링 하는 방법이다. 예를 들어, 그림 9(a)의 경우, DM으로 모델링 된 FSM의 state를 reduction ratio로 평균화된 state로 만들 수 있다. 그림 9(b)의 경우에는, 불규칙한 동작을 가지는 burst type을 확률적으로 모델링 하는 경우로 실제 random한 동작을 가지는 신호를 모델링 하기에 적합한 방식이다.



(a) Bus request interval의 NDM 예
 (b) Burst type의 NDM 예
 그림 9. BFM output 모델링 방식: *Non-Deterministic*

Trace-Driven Method(TDM)는 그림 10과 같이, 모니터 결과를 BFM에서 순차적으로 읽고 내보내는 방법으로서, event나 RTC와 같은 transaction control이 요구되지 않는 경우의 동작 모델링에 적용하면, 실제 RTL과 동일한 동작을 가지는 모델을 구현할 수가 있다. 참고로 당사의 DSP processor인 CalmRISC의 경우에는 cache 시뮬레이션 결과를 tracing하여 AHB transaction으로 변환시켜 모델링을 수행 하였다.

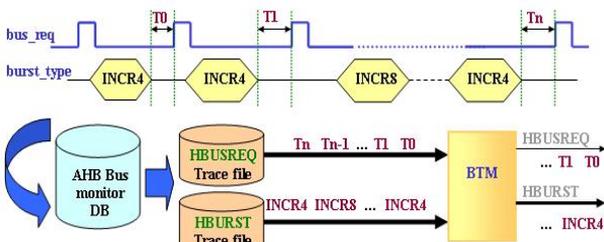
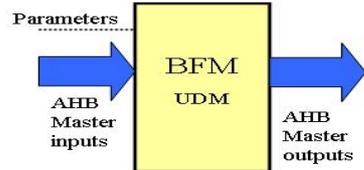


그림 10. BFM output 모델링 방식: *Trace-Driven*

User-Defined Method(UDM)는 ARM processor와 같이 칩 레벨 시뮬레이션에서 dynamic한 동작이 발생하는 경우, IP의 bandwidth를 계산하여 평균적인 transaction interval과 transaction control 신호를 발생하도록 parameter set을 변화시켜 BFM을 생성하는 방법이다(그림 11). UDM은 BFM output control 신호의 sequence와 event, RTC와 같은 control 동작을 고려하지 못하기 때문에 model의 정확도는 낮아지지만, parameter를 설정하는 방법으로 쉽게 모델링이 가능하다는 장점이 있다. 특히, 플랫폼에서 virtual IP를 plug하여 성능평가를 하기 위한 환경에는 UDM에 의한 모델링 방법이 효과적이다.



Parameters: req_wait_cycle, total_cycle, r_burst_type, r_addr_start ...

그림 11. BFM output 모델링 방식: *User-Defined*

위에서 설명한 BFM 모델링 방법론을 이용하여 video processor (VP)의 transaction 동작을 design configuration file에서 기술하는 방법에 관한 예제를 그림 12에 나타내었다.

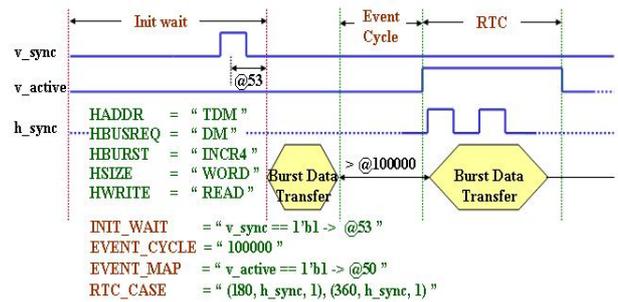


그림 12. Design configuration 기술 예제

3. 플랫폼 구축

DVD player SoC 플랫폼상의 모든 AHB master 들을 BFM으로 모델링 하여 기존의 RTL-base에서 BFM-base로 플랫폼을 재구성하였다. 표 1은 각 AHB master들의 BFM 모델링 방법을 나타내고 있으며, BFM 모델의 정확도를 높이기 위하여 BFM output별로 IP의 특성에 맞게 각각 4개의 모델링 방법을 조합적으로 선택하여 모델링 하였다. 총 15개 중 6개의 IP는 INIT_WAIT, EVENT, RTC_CASE 등의 BFM control 이 고려되었다.

표 1. 각 AHB master의 BFM 모델링 방법

ID	IP	Modeling 방법						
		BTM outputs				BTM Control		
		HBUSREQ	HWRITE	HSIZE	HBURST	Init wait	Event cycle	RTC
1, 8	FEU	TDM	TDM	TDM	TDM	En	-	-
2	IODMA	DM	TDM	WORD	INCR4	ch_req	고려	-
3	SDIN	DM	READ	WORD	INCR4	sd_ydata	고려	고려
4, 5	VP	DM	READ	WORD	INCR4	h_sync	고려	고려
6	MIXER	DM	READ	WORD	TDM	h_sync	고려	고려
7	SPD	TDM	READ	WORD	TDM	pbx_req	-	-
9	CalmADM	Cache simulation 결과를 BTM으로 자동변환				En	고려	고려
10	ARM940T	User-defined model				En	-	-
11	PSD	NDM	NDM	NDM	NDM	En	-	-
12, 13	MEVD	NDM	NDM	WORD	NDM	En	-	-
14	IFO	TDM	TDM	WORD	INCR4	En	-	-
15	GA	TDM	TDM	WORD	TDM	En	-	-

4. 성능 측정 모니터

BFM-based 플랫폼을 구성한 후, AHB bus 레벨에서 transaction latency와 RTC의 성능을 평가하기 위하여 Verilog 모니터와 분석 프로그램을 개발하였다(그림 13).

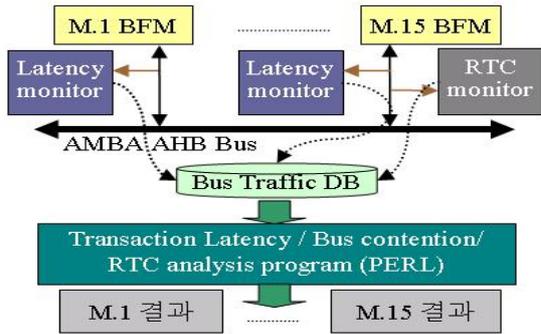
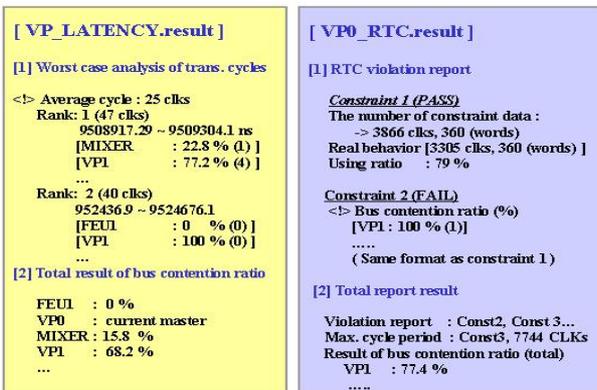


그림 13. 성능(Transaction Latency, RTC) 분석 환경

Transaction latency는 그림 14(a)와 같이 각 BFM의 bus arbitration/transaction delay의 평균값과 worst 값을 bus contention ratio 정보와 함께 분석되므로 critical delay case가 성능에 미치는 영향을 확인할 수 있다. 이러한 분석 결과로, bus layer 구조와 arbitration scheme을 평가할 수 있다.

RTC 분석은 그림 14(b)와 같이, 각 IP별로 정해진 시간 내에 전송해야 할 data 양을 체크 하는 방법으로 이루어진다. Worst RTC violation이 발생하는 경우의 전송 delay를 분석 하여 buffer size를 최소화할 수 있으며 bus contention ratio 정보도 같이 얻을 수 있다.



(a) Latency 분석 결과 (b) RTC 분석 결과
그림 14. 성능(Latency, RTC) 분석 결과 예

III. 실험 결과

DVDP 플랫폼에서 RTL master의 communication 동작을 BFM-based 플랫폼으로 구성하였다(그림 15). BFM을 이용한 top-level 시뮬레이션은 표 2와 같이 각 IP별로 hard bus traffic이 발생하는 동작 모드의 경우를 각각 모델링 하여 worst-case 시나리오를 설정하였다.

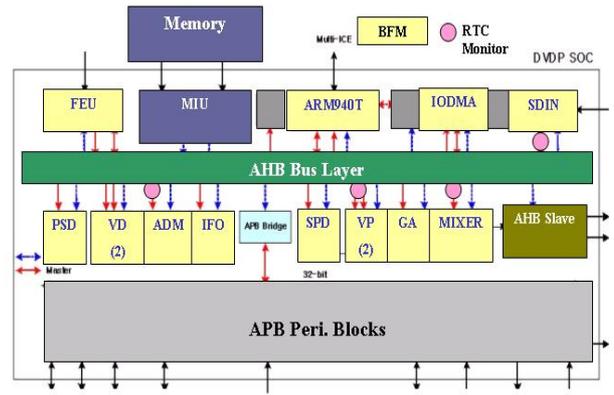


그림 15. BFM-based DVDP 플랫폼

RTL과 BFM 각각의 top-level 시뮬레이션 결과에서, MIXER, Video processor(VP0, VP1), 그리고 SPD의 transaction accuracy를 성능측정 모니터를 이용하여 분석하였다. 그림 16에서, 막대 그래프는 왼쪽부터 RTL/BFM bus transaction delay와 RTL/BFM worst transaction delay를 비교한 결과를 반영한 것으로, 4개 IP의 BFM 모두 RTL 대비 오차율이 10% 이내로 BFM의 높은 정확도를 검증할 수 있었다.

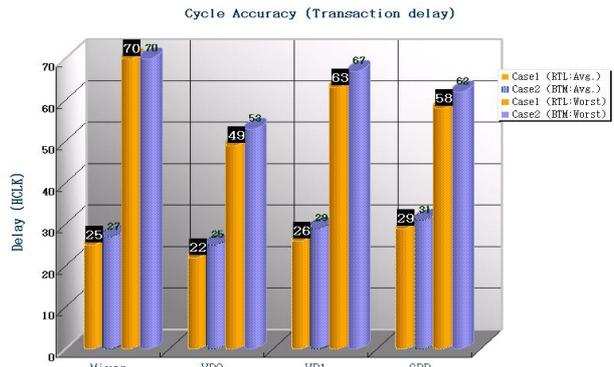


그림 16. RTL 대비 BFM의 transaction 정확도 비교

표 2는 각 BFM의 latency (arbitration/transaction delay)의 분석 결과이다. 성능평가 환경에서 worst arbitration delay의 경우, 각 BFM간의 bus contention ratio도 함께 분석이 가능하므로 latency의 최적화를 위한 multi-layer bus 구조나 bus access priority의 조절을 위한 정보로 이용할 수가 있었다. 본 실험에서 BFM간의 bus contention ratio는 가장 높은 경우가 최대 20% 범위 내에 있었으며, 다른 BFM간의 contention ratio도 비교적 평균화 되어 있어, arbitration scheme이 우수하다는 것을 결과에서 확인할 수 있었다.

IV. 결론

표2.Top-level 성능 (latency, bus contention) 분석 결과

ID	IP	Operation Mode	Worst Latency		
			Avg. Arbitration Delay (Worst) [HCLKs]	Avg. Transaction Delay (Worst) [HCLKs]	Bus contention Ratio [%]
1,8	FEU(0,1)	DVD_IX_PLAY	37(809), 39(312)	42(317), 42(315)	18.0
2	IODMA	WORST	25(366)	27(388)	15.0
3	SDIN	4/1_IMAGE	28(168)	33(173)	20.0
4,5	VP(0, 1)	3D-IPC	31(165), 31(175)	35(169), 35(183)	18.0
6	MIXER	SBPP	35(385)	43(389)	15.0
7	SPD	NORMAL	61(439)	72(450)	15.0
9	CalmADM	AC3_DECODE	41(728)	48(740)	18.0
10	ARM940T	NORMAL	34(1049)	39(1053)	18.0
11	PSD	DESC_16	42(1044)	46(1048)	16.0
12,13	M2VD(0, 1)	ICEWALL	68(1056), 61(1066)	72(1061), 66(1071)	15.0
14	IFO	FITOP_16	68(1091)	72(1096)	13.0
15	GA	SBPP_BITBLT	90(1069)	93(1072)	20.0

본 실험 결과에서 *INIT_WAIT*, *EVENT_MAP*, *RTC_CASE*와 같은 BFM control 방식을 고려한 모델링 방법이 실제 RTL과 같은 동작을 모델링 할 수 있는 것을 확인했다. 예를 들어, 그림 17은 VP BFM이 RTC violation이 발생여부를 시뮬레이션 과정에서 판단하여 burst transaction sequence를 control하고 있는 동작을 보여주고 있다.

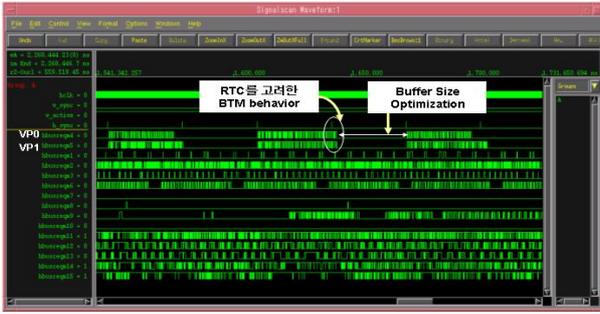


그림 17. RTC 동작이 구현된 VP BFM

한편, throughput 관점에서 VP0, VP1, MIXER의 3개 BFM만을 대상으로 RTC를 분석하였고, 그 결과는 표 3에 나타내었다. MIXER에서는 RTC violation이 발생하는 경우가 없었으며, VP0와 VP1의 경우에는 약 10%의 확률로 RTC violation이 발생하였다.

RTL VP의 경우에는 RTC violation이 발생하지 않도록 1 horizontal sync까지의 data transaction delay(약 3720 bus 클럭)를 buffer로 충당하고 있다. 하지만, 본 실험에서 worst RTC latency는 constraint cycle인 3720 클럭 대비 평균 5%이내의 범위로서, 실험 결과를 분석하여 buffer size의 최적화를 제시하였다.

표 3. Top-level throughput(RTC) 분석 결과

IP	RTC			
	# Violations	Constraint Cycle [HCLKs]	Worst Latency [HCLKs]	Average Latency [HCLKs]
VP(VP0)	38	3720	3830	3284
VP(VP1)	33	3720	3942	3323
MIXER	0	3720	3436	3231

본 논문에서는 RTL 대비 높은 정확도를 가지면서 고속으로 시뮬레이션 할 수 있는 모델을 개발하는 방법을 제안 하였으며, 현재 모든 flow는 자동화 되어 있다. RTL의 communication 부분은 다양한 테스트 시나리오에서 서로 다른 bus traffic 상황에 따라 달라지는 transaction 동작을 최대한 고려하여 높은 정확도의 모델을 개발 하기 위한 BFM control 방법론(*INIT_WAIT*, *EVENT_MAP*, *RTC*)과 BFM output 신호의 모델링 방법론(DM, NDM, TDM, UDM)을 적용하였다. 본 실험에서는 RTL 대비 BFM의 transaction 동작이 90% 이상의 높은 정확도를 보였으며, 전체 시뮬레이션 시간이 1시간 미만으로 고속으로 아키텍처를 평가하는 것이 가능하였다.

한편, 성능 평가 flow를 개발하여, BFM-based top-level 시뮬레이션을 실행하였고, 각 BFM의 average (worst) bus latency, throughput(RTC)을 bus contention ratio와 함께 정량적으로 성능 분석 하였다. 이러한 성능 분석 환경으로 bus arbitration scheme과 bus layer 설정 그리고 IP의 internal buffer size를 최적화 할 수 있도록 하였다.

감사의 글

본 논문의 작성에 있어, 기술적인 많은 조언과 실험환경을 제공해 주신 같은 사업부 소속의 Media Optical P/J팀의 모든 분에게 감사의 말씀을 드린다.

참고문헌

- [1] L. A. Cortes, P. Eles, Z. Peng, “ Hierarchical Modeling and Verification of Embedded Systems” , Euromicro Symposium on Digital Systems Design, 2001.
- [2] G. Varatkar, R. Marculescu, “ Traffic Analysis for On-chip Networks Design for Multimedia Applications” , IEEE DAC, 2002.
- [3] K. Lahiri, A. Raghunathan, S. Dey, “ System-Level Performance Analysis for Designing On-Chip Communication Architectures” , IEEE TCAD, 2001.
- [4] Gill, P., Richard Hersemeule, Gaston Cambon, “ Bus analysis and performance evaluation on a SOC platform at the system level design” , VLSI SOC, 2001.
- [5] K. Richter and R. Ernst, “ Event model interfaces for heterogeneous system analysis” , DATE, 2002