

Design Space Exploration of IEEE 802.11n using SystemC

Sung-Rok Yoon, Jin Lee, and Sin-Chong Park
Bit Engineering Lab, System Integration Technology Institute
Information and Communications University
119, Munjiro, Yuseongu, Daejeon
Korea

Abstract: - This paper proposes the case study of the design space exploration using SystemC. The platform based design with BUS based communication architecture is used as designing approach. IEEE 802.11n system is used as case application. Transaction level modeling(TLM) of the platform is generated using SystemC. Various architectural options – CPU and BUS clock speed, memory response time, memory size -- are evaluated by the application specific performance metrics. The architecture is verified by the real-time constraint of IEEE 802.11n. Further evaluation of architecture is performed with the additional performance metrics in terms of the average and instantaneous throughput of uplink traffic.

Key-Words: - Transaction Level Modeling, SystemC

1 Introduction

The growth of integration technology introduces the great demands on high-performance applications. Unfortunately, designer's skill is hard to follow such demands and this productivity gap becomes one of main issues in SoC design.

Platform-based design with IP reuse and design decomposition becomes one of SoC design trend to reduce time to market, designing cost, and at last overcome the productivity gap. However, it contains many SoC components such as central processing units, coprocessors, HW components, memories, I/O peripherals, and communication architectures. The methodology to evaluate many candidates of platform in early stage with small effort is required [1].

There have been many researches on abstraction level and simulation. Transaction level modeling(TLM) using SystemC is one of main trends. Transaction level is intermediate abstraction level between behavior and RTL[1]. SystemC is open-source language which provides useful C++ libraries to model system architecture with various abstraction levels. The standard for SystemC is approved in 2005, and it is expected that the language will be more commonly used[2].

In this paper, high-level architecture exploration methodology is proposed with practical example. The paper adopts high-end wireless application, IEEE 802.11n system, as the case example. The real-time requirement and the throughput of the application are evaluated in order to verify the established platform is

well designed. Also, several architectural parameters are changed to refine the architecture.

There have been many works related with the methodologies of system level modeling with SystemC. Authors in [9] provides the generic framework to generate TLM of Multi-Processor SoC(MPSoC) platform. Their method is well-defined and applicable to generate TLM for many SoC components – single and multiple threaded processing element, BUS oriented communication architecture, NoC, and so on. But it has avoidable context switches or events which slow down the simulation speed as they are not optimized for the certain architecture. TLM for general SoC components -- AMBA AHB BUS, SPRAM, DPRAM -- should be optimized in terms of speed and accuracy as they can be reused without modification after the first creation if their interface is well defined. This work generates optimized TLM for several SoC components. They are reusable as they follows generic TLM interfaces which are strong draft for TLM standard library.

There is lack of case studies on TLM. This work provides one working solution for IEEE 802.11n system, which has not been explored using TLM approach. TLM of other applications are presented in [5, 8].

The remained part of this paper is organized as follows: In section 2, IEEE 802.11n system and its real-time constraint are introduced. In section 3, the architectural analysis under real-time constraint is performed. The first architecture is extracted from the

analysis. In section 4, the architectural refinement considering more performance metric and additional simulation is performed. The simulation environment and result are illustrated. Our conclusion will be drawn in section V.

2 IEEE 802.11n system

IEEE 802.11n is the next generation wireless LAN standard whose main objective is the enhancement in throughput on both MAC and PHY layer[3, 4]. In MAC layer, it defines new efficient data transmission strategy such as frame aggregation scheme, which aggregates MPDU(MAC protocol data unit)s up to 65535 bytes to generate single PSDU(PHY service data unit)[3]. This scheme is effective for the case of high PHY rate transmission because the additional delay which comes from MAC protocol(back-off, inter-frame spaces, header) becomes more dominant in the case. In addition, IEEE 802.11n optionally allows the reverse link transmission within a TXOP(transmission opportunity[3]. In PHY layer, physical rate should be achieved to 600 Mbps using MIMO(Multi-Input Multi-Output) technique and expanding channel bandwidth[4].

IEEE 802.11n system has the real-time constraint which comes from SIFS(short inter frame space) as shown in fig. 1. It is an interval between receiving data frame and responding acknowledgement frame in down-link process. 'RX_ACK_delay' should be always less than 'Real_time_constraint' in the figure.

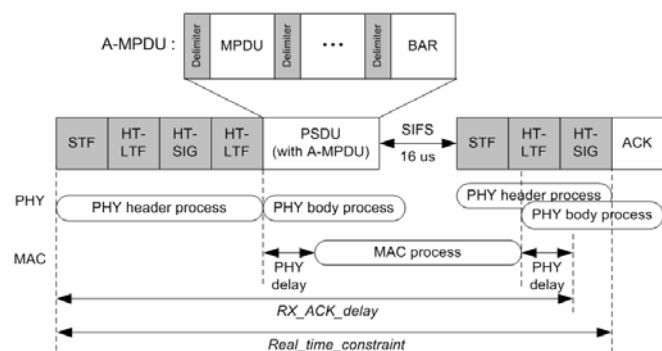


Fig. 1 Real-time constraint of IEEE 802.11n system during down link process (BAR: Block ACK request, ACK : Acknowledgement)

Above constraint is the basic consideration in this work. In addition, the work focus on the MPDU aggregation and reverse link transmissions. EDCA in IEEE 802.11e standard is used as channel accessing mechanism.

3 Architecture Analysis

The objective of high-level TLM in this work is the estimation of the system performance with less effort. Platform is modeled as abstracted form. Fig. 2 shows the platform. A single CPU is used for MAC processing and a dedicated HW core is used for HW. SW MAC has benefits as the new MAC protocols are downloaded after implementation and the research on MAC protocols are consistent than PHY.

Dual-port RAM is used to synchronize the communication between MAC and PHY layer. PCMCIA is used to communicate with upper layer(LLC). AMBA AHB is used as the communication architecture. Table I describes the architectural parameters considered.

TABLE I
ARCHITECTURAL PARAMETERS

Symbol	Unit	Description
C_{clock}	Hz	CPU clock speed
B_{clock}	Hz	BUS clock speed
B_{width}	bit	BUS width
DP_{acc}	sec	Access time of DP-RAM
DPT_{size}	Byte	Dual-port RAM size for TX
BU_{max}	number	Maximum size of burst transfer using BUS

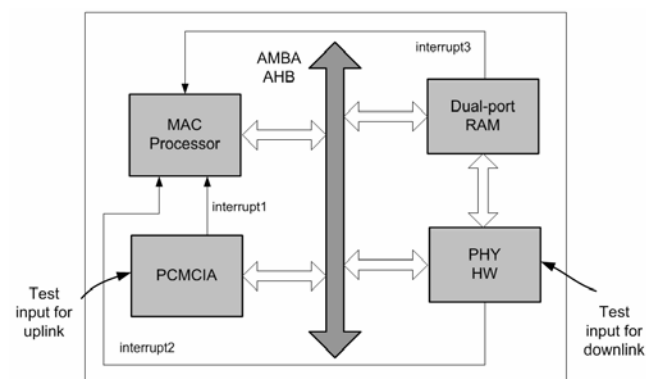


Fig. 2 Simple block diagram of TLM platform

The architectural parameters are decided by considering the real-time requirement. Downlink packet with the maximum size(65535 byte), aggregation number(64), and physical rate(600Mbps) is applied as the corner case input.

From fig. 1, the requirement can be expressed as follow :

$$RX_ACK_delay \leq Real_time_constraint \quad (1)$$

$$RX_ACK_delay = PHDR_{dur,RX} + PHY_{lat,RX} + MAC_{pre,RX} + PHY_{lat,TX} \quad (2)$$

$PHDR_{dur,RX}$ is the PHY header duration which includes the duration for STF, HT-LTF, and HT-SIG in fig. 1. The duration ranges from 20 to 40 us, and

depends on MCS(modulation and coding scheme) and PPDU frame type. $PHY_{lat,RX}$ and $PHY_{lat,TX}$ are the latencies for transmission and reception on PHY layer. $MAC_{proc,RX}$ is the processing delay for MAC layer to get the data from received frame and to prepare ACK for it. It can be calculated as follow:

$$MAC_{proc,RX} = \sum_{i=1}^{A_{num}} \left(P_{i,DEL} + P_{i,HDR} + \sum_{N_{MSDU(i)}} P_{i,CRC} \right) + P_{i,BA} \quad (3)$$

where $P_{i,job} = R_{i,job} + \frac{C_{cyc,job}}{C_{clock}} + W_{i,job}$

$P_{i,job}$ is the processing delay to process ‘job’. ‘DEL’ is the delimiter field in A-MPDU frame(see fig. 1), and ‘HDR’ is a MPDU header. ‘CRC’ and ‘BA’ are the cycle redundancy check and the block ACK respectively. $P_{i,job}$ is divided into three phases—read, process, and write. In ‘read’ phase, required data to process the job is moved to the cache. It is processed by CPU in ‘process’ phase and moved to the system memory or other components in ‘write’ phase. $C_{cyc,job}$ is the number of cycles to process ‘job’. $R_{i,job}$, and $W_{i,job}$ are the communication delay, and can be expressed as following when BUS is used.

$$R_{i,job} = \left\lceil \frac{R_{num,job}}{BU_{max}} \right\rceil \frac{B_{cycle,ARB}}{B_{clock}} + \lceil S_{acc} B_{clock} \rceil \frac{R_{num,job}}{B_{clock}} \quad (4)$$

$R_{num,job}$ is the number of transfers to process ‘job’. $B_{burst,max}$ is the maximum number of the burst transfer, and $B_{cycle,ARB}$ is the number of cycles used for the arbitration in BUS. S_{acc} is the access time of slave, and it is same as DP_{acc} for the dual-port RAM.

The computation delays for jobs are shown in table II. They are measured by simulation or benchmarked by white papers in public domain. As a result, the graph in fig.3 is obtained. The result uses ‘BW32-BU1-BF100-M20’ as default configuration, and changes each architectural parameter while others are fixed. Changing BUS width and the burst transmission requires less than 650 MHz clock speed to achieve the requirement. As a result, the architectural candidates are obtained.

TABLE II
FIXED PARAMETERS

Symbol	Value	Symbol	Value
$C_{cyc,DEL}$	104 cycle	$C_{cyc,HDR}$	142 cycle
$C_{cyc,CRC}$	224 cycle	$C_{cyc,BA}$	98 cycle
$PHDR_{dur,RX}$	36 us	$PHY_{lat,RX}$	10 us
		$PHY_{lat,TX}$	

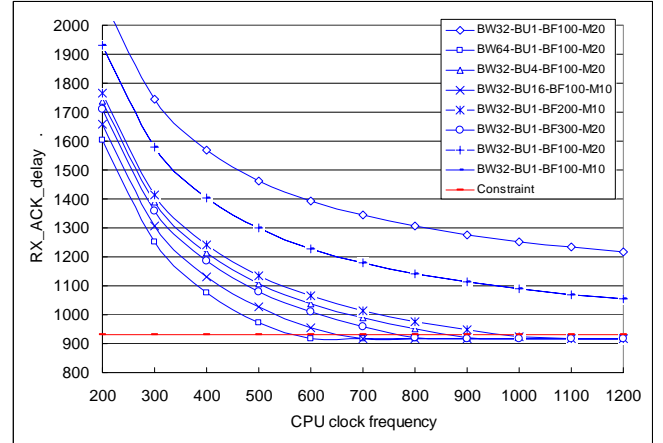


Fig. 3 Comparison of architectural options in terms of the real-time constraint (BW : B_{width}, BU : BU_{max}, BF : B_{cyc}, M : DP_{acc})

4 Architecture Refinement for High Performance

In order to evaluate the performance of WLAN system, there is the uplink throughput of a station to be considered. Although there is no requirement to keep the MAC protocol, the uplink throughput is closely related with QoS.

The performance of the system(THR_{SYS}) is not the unique parameter for the uplink throughput(THR_{UL}). Input traffic load(THR_{IN}) and the channel throughput ($THR_{CHANNEL}$) restrict the performance.

$$THR_{UL} = \min(THR_{SYS}, THR_{CHANNEL}, THR_{IN}) \quad (5)$$

The average of THR_{IN} does not exceed 100 Mbps in usual case. However, the throughput during certain period or instance is required to have more than 100 Mbps because the station can transmit its data during the instantaneous interval. For example, 1 Gbps is required as for the average 100 Mbps uplink throughput when 1/10 of bandwidth is allocated for the station.

The random traffic model is shown as Table III. In order to simplify the observation which is considered with various architectural options, TXOP is assigned only for downlink. The TLM station has limited chances to transmit data using reverse-link with Block-ACK and MPDU aggregation as Fig. 4 shows. A-MPDU has a random size by the state of transmission buffer. TXOP is assigned to 3ms, but it is naturally randomized by stopping when there remains

no data to transmit within TXOP. Physical rate is fixed to maximum value, 600 Mbps.

TABLE III
RANDOM TRAFFIC MODEL

Symbol	TLM station	AP
Offered Load		100 Mbps
Chance to get TXOP	0%	33%
Average MSDU gen. time	80 us (poisson process)	
MSDU length	Uniform : [500~1500 byte]	

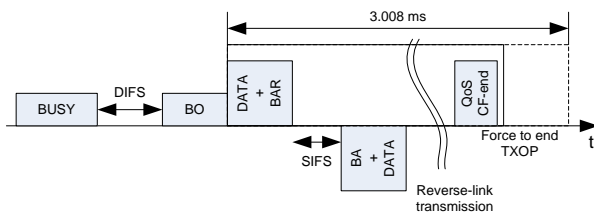


Fig. 4 Transmitting uplink data using reverse-link in TXOP

TLM of platform in fig. 2 is developed using SystemC. For the master to access the AHB or for the AHB to access the slaves, the bidirectional blocking interface(tlm_transport_if) in TLM draft library is used. Interface is implemented in AHB and slaves.

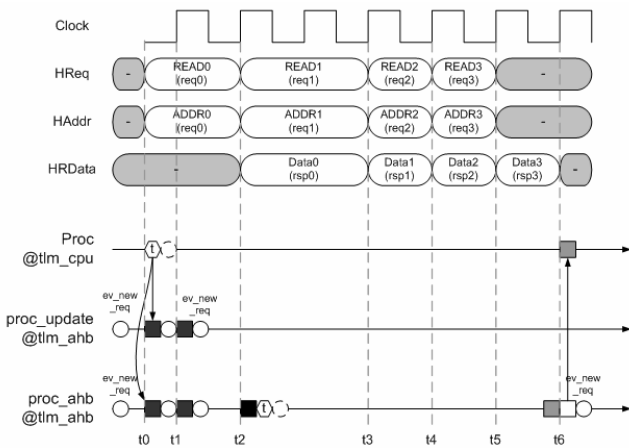


Fig. 5 TLM of AMBA AHB signaling protocol (circle : start of event waiting, dark square : end of event waiting, white square : event notification, hexagon : call of tlm_transport_if method)

Fig. 5 shows the mechanism of time annotation in TLM of AHB. CPU tries to access to AHB at t0 by calling tlm_transport_if method. It is blocked until it receives the four read data at t6. In AHB, proc_update receives the requests from the masters and updates the number of pended request. proc_ahb manages AHB operation. It identifies requests at t1 and delays

additional one clock for the address phase. Then, it access to the slaves by calling tlm_transport_if method and gets four read data at t6. Data is returned to CPU at the same time.

As the platform uses a single CPU for MAC processing, the CPU has multiple threads for uplink and downlink processes and it should solve the confliction between them. Fig. 6 shows the abstracted operation of the CPU. Using “sc_mutex” channel in SystemC, only one of threads occupies CPU and process their job.

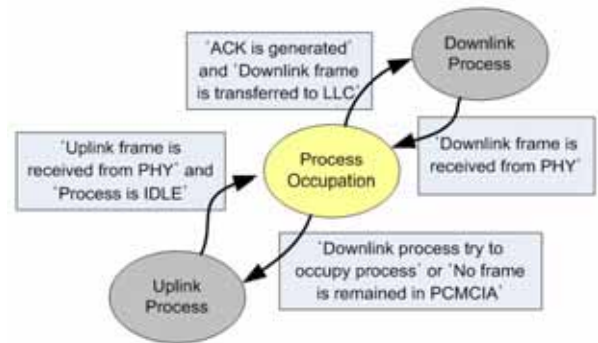


Fig. 6 CPU modeling (uplink process checks the trial of process occupation by downlink process periodically)

Fig. 7 shows the average throughput during observation period. Four architectural parameters— C_{clock} , B_{clock} , DP_{acc} , and DPT_{size} —are changed. Uplink throughput of station is affected by architecture configurations. For example, 30% increment of CPU speed leads to 20% increment for ‘33K-UL’ and 10% increment for ‘17K-UL.’ Also, the twice increment of BUS speed leads 40% and 100% increment respectively. When TX buffer size is large(65Kbyte in this case), there is no difference by architectural changes. It is not required to be processed when transmission opportunity is given repeatedly. The sum of uplink and downlink traffic is maintained to 200Mbps for every case.

Fig. 8 shows the instantaneous throughput of uplink traffic. When the diamond graph is compared to others, the throughput is maintained during TXOP because it has more data to be prepared. When the triangle graph and square graph are compared, the enhancement in BUS clock speed or memory access time leads to maintain the uplink throughput.

The simulated case is valuable to design AP as it distributes data to multiple stations with limited channel accessing chance. The reverse-link transmission allows AP to send its data to the station which has the current right to access channel. It can

enhance QoS because data transmission chance of AP will be well-distributed and it can help to reduce drop-rate.

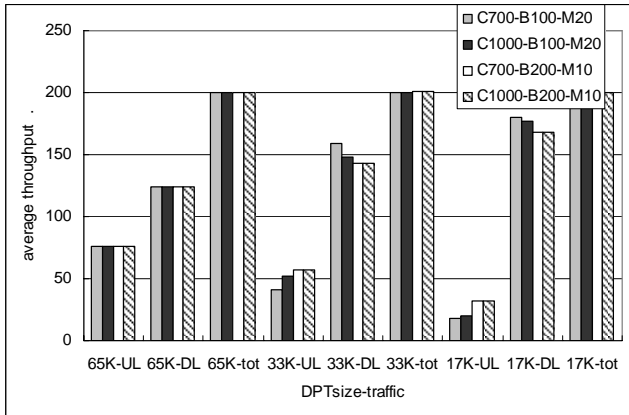


Fig. 7 Average throughput (DPT_{size} = 65000, 33000, 17000; UL : uplink, DL : downlink, tot : total)

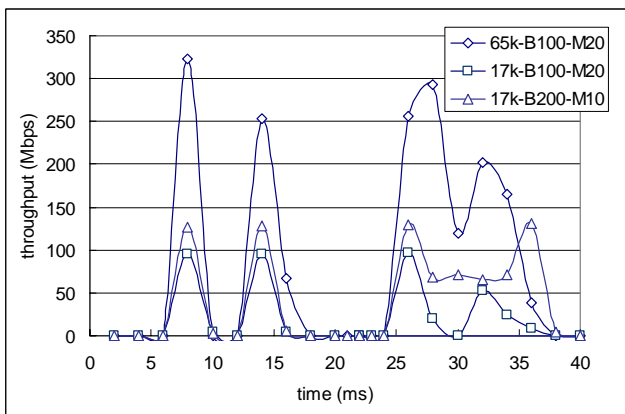


Fig. 8 Instantaneous throughput for uplink observed by 2 ms period

5 Conclusion

The paper proposes the case study of high level TLM with the practical application. IEEE 802.11n system is applied to the BUS based SoC platform and its various architectural options are evaluated. Simple analysis considering the real-time constraint is done for the initial parameter setting. Then, the TLM platform is developed for the further evaluation and refinement of architecture in terms of the average and instantaneous throughput of uplink traffic.

As the first stage TLM simulation, lots of architectural parameters, especially related with computation, are abstracted and fixed. However, it has capability to refine further in terms of accuracy. Transactor which translates TLM transaction to RTL and vice versa generates unified verification

environment. With this concept, several candidates extracted from the first simulation in this work will be evaluated with less simulation speed but higher precision level.

Acknowledgements:

This work was partly sponsored by ETRI SoC Industry Promotion Center, Human Resource Development Project for IT SoC Architect.

References:

- [1] Frank Ghenassia, Transaction Level Modeling with SystemC, Springer, 2005.
- [2] IEEE Standard SystemC Language Reference Manual, 2006.
- [3] Joint Proposal: High throughput extension to the 802.11 Standard: MAC, IEEE, 2006.
- [4] Joint Proposal: High throughput extension to the 802.11 Standard: PHY, IEEE, 2006.
- [5] M. Bonaciu, et al. "High-Level Architecture Exploration for MPEG4 Encoder with Custom Parameters," Proceedings of ASP-DAC, 2006.
- [6] ANSI/IEEE Std 802.11, 1999.
- [7] A. Rose, et al. "Transaction Level Modeling in SystemC," <http://www.systemc.org>
- [8] Jin Lee and Sin-Chong Park, "Methodology of High-Level Transaction Level Modeling using 802.11 PHY Example," IEICE Trans. on Communication, Vol.E88-D, No.7, pp.1749-1753, July, 2005.
- [9] Tim Kogel, et al., Integrated System-Level Modeling of Network-on-Chip enabled Multi-Processor Platforms, Springer, 2006.