Interconnect Delay Aware RTL Verilog Bus Architecture Generation for an SoC

Kyeong Ryu, Alexandru Talpasanu,
Vincent Mooney and Jeffrey Davis
School of Electrical and Computer Engineering
Georgia Institute of Technology

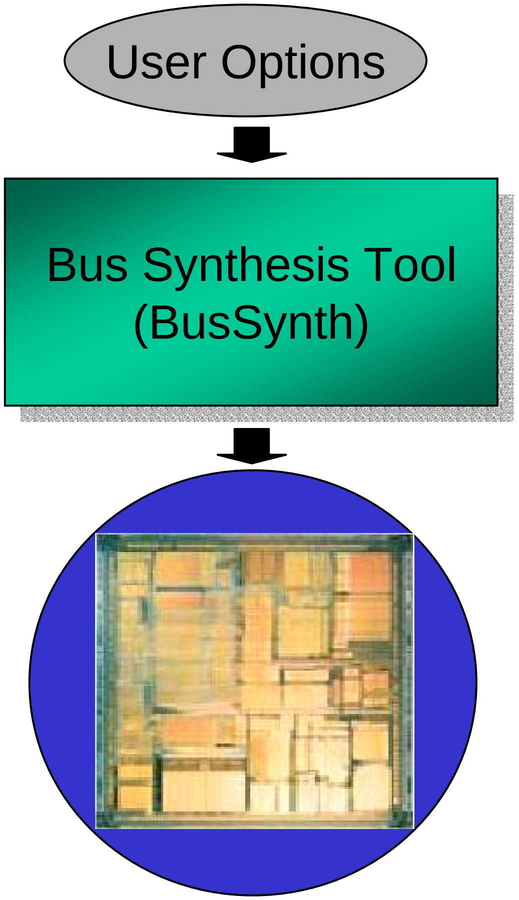
August 2004

Outline

- Introduction
- Interconnect Delay Estimation
- Interconnect Aware Module Generation
- BusSynth Overview
- Application Example
- Conclusion

Introduction

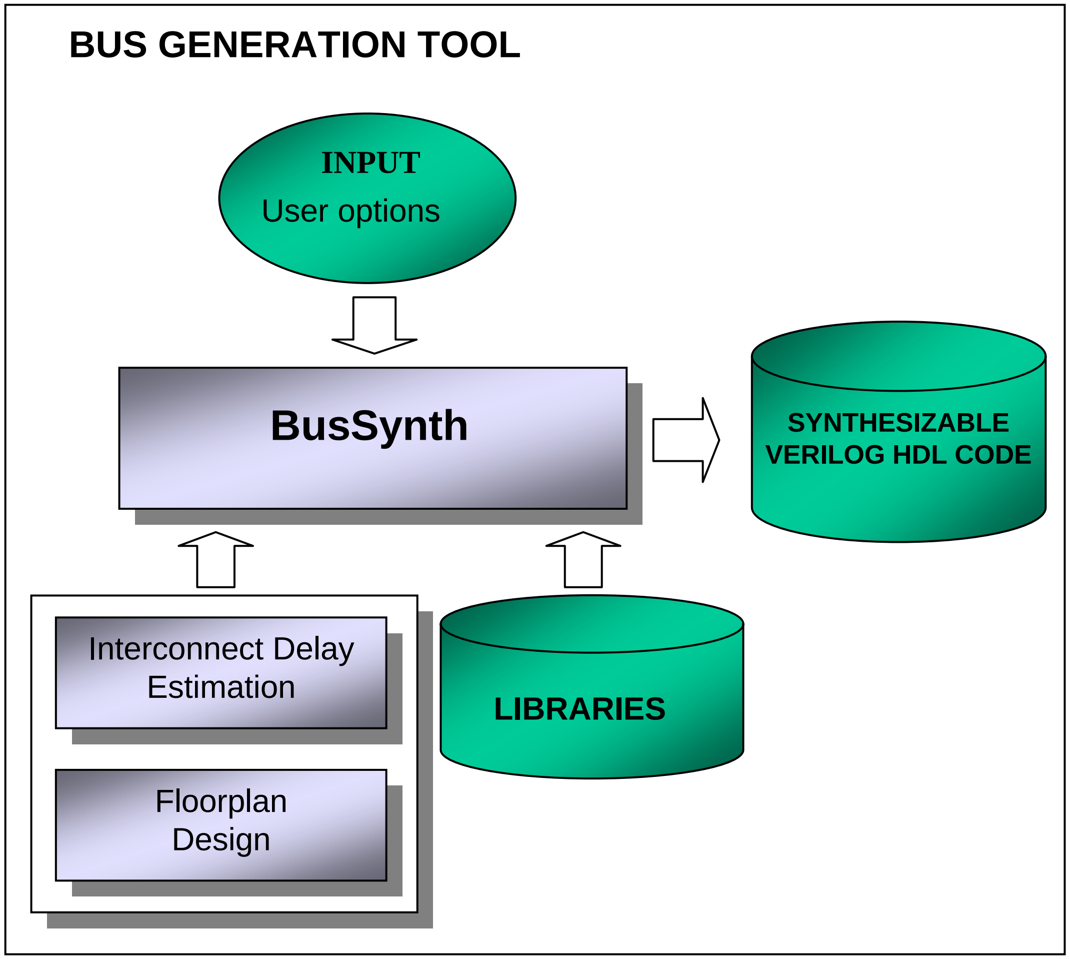
- A methodology to generate a custom bus architecture using accurate estimations of interconnect delay
  - Easy and quick design of an SoC bus system
  - Fast design space exploration across performance influencing factors
  - Development of a bus synthesis tool (BusSynth)
  - Register-transfer level HDL output based on user options and interconnect delay



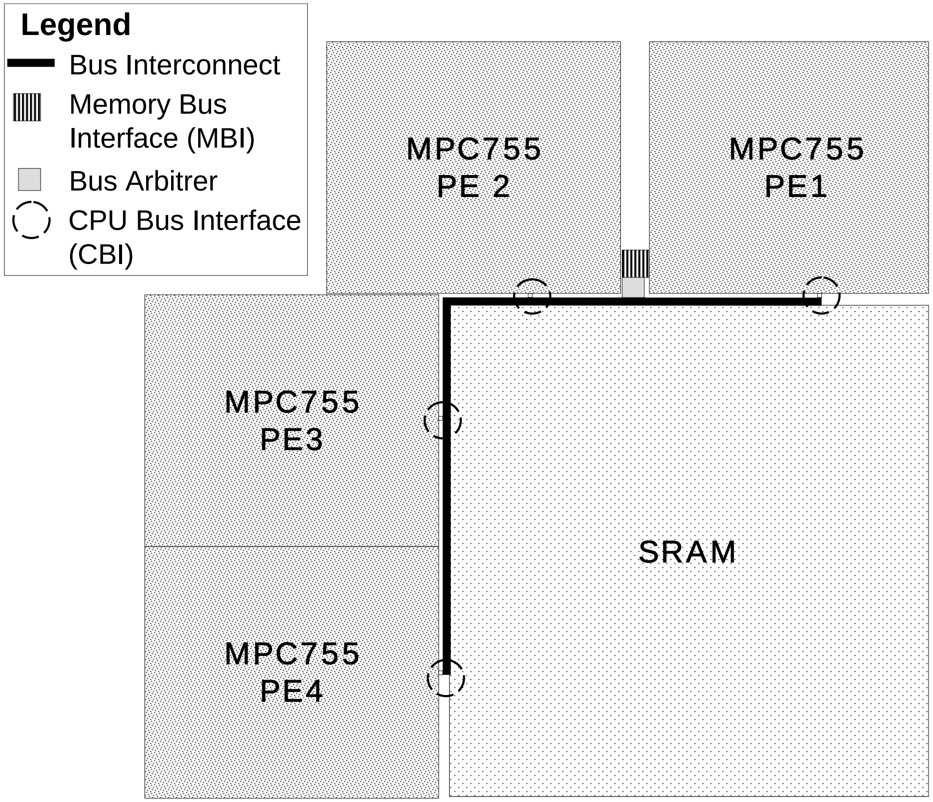
Related Work

- Shin *et al*. ('04), "Fast Exploration of Parameterized Bus Architecture for Communication-Centric SoC Design" [5]
  - A single type of bus topology
- Thepayasuwan *et al.* ('04), "Layout Conscious Bus Architecture Synthesis for Deep Submicron Systems on Chip" [6]
  - A single type of bus topology
- BusSynth
  - A variety of bus types including multiple and heterogeneous type
  - Interconnect delay aware bus generation

Bus Synthesis (BusSynth) Overview



Interconnect Length Estimation



(a) Estimated Floorplan

	SRAM Memory	
	Length [cm]	**Delay [ns]**
Processing Element 1	0.2521	0.2848
Processing Element 2	0.6143	0.5727
Processing Element 3	1.2753	2.2882
Processing Element 4	1.9363	3.0472

** TSMC 0.25 μm Design Rules

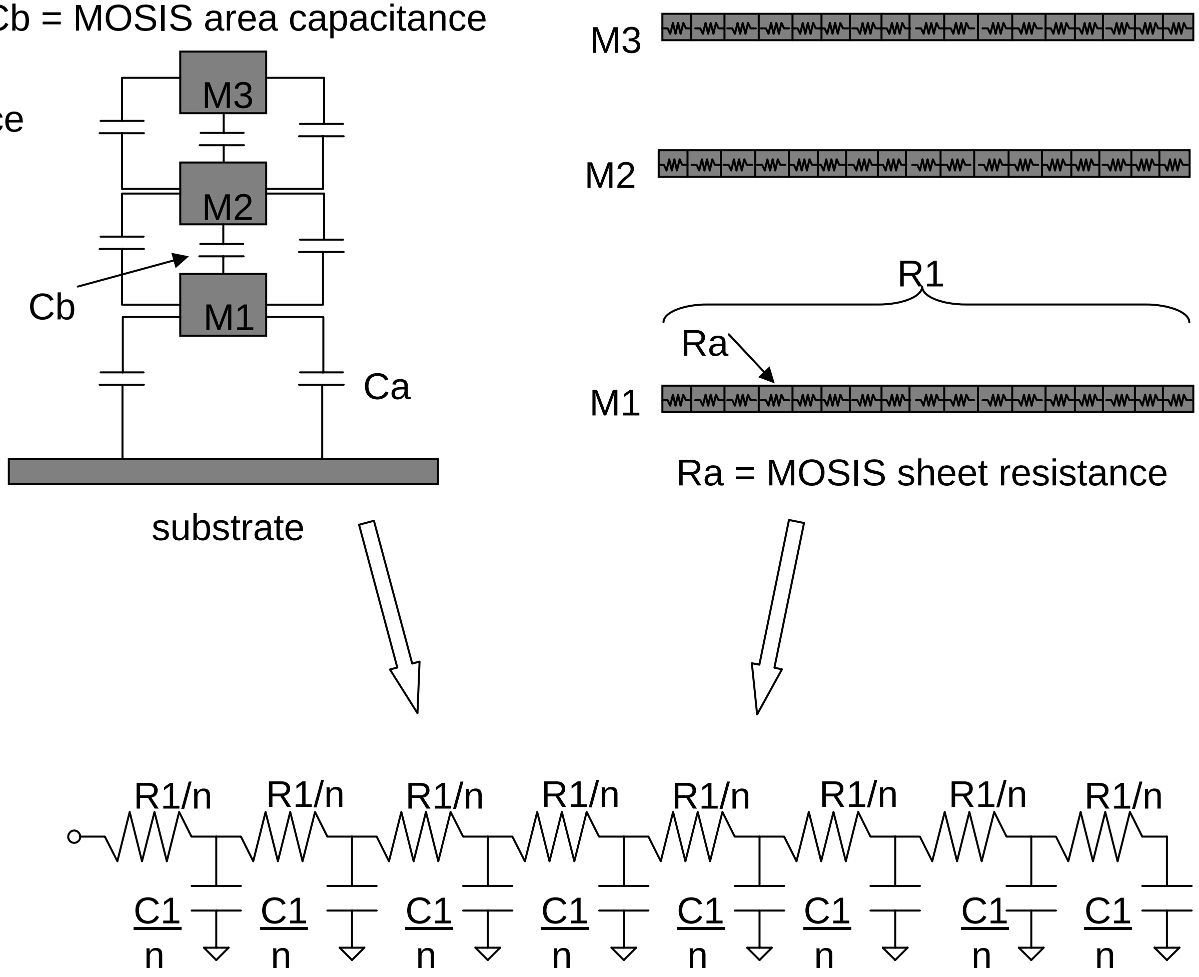
(b) Interconnect length estimation

Interconnect Model Parameters

Ca = MOSIS fringe capacitance

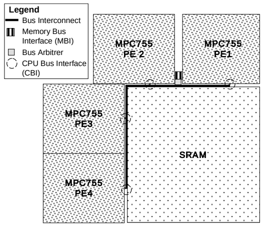
Cb = MOSIS area capacitance

Coupling capacitance effects explained in technical report [11]



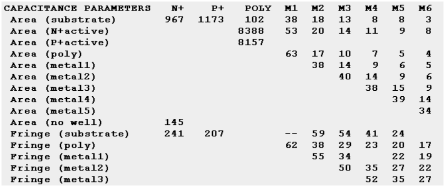
Ra = MOSIS sheet resistance

Accurate Interconnect Delay Estimation



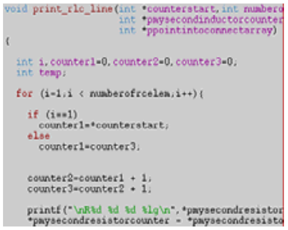
Floorplan Bus Interconnect Length Calculation

MOSIS Process Parameters



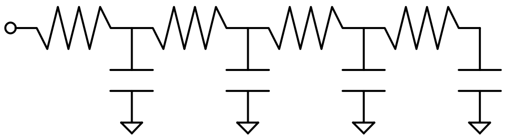
CAPACITANCE PARAMETERS	N+	P+	POLY	M1	M2	M3	M4	M5	M6
Area (substrate)	967	1173	102	38	18	13	8	8	3
Area (N+active)			8388	53	20	14	11	9	8
Area (P+active)			8157						
Area (poly)				63	17	10	7	5	4
Area (metal1)					38	14	9	6	5
Area (metal2)						40	14	9	6
Area (metal3)							38	15	9
Area (metal4)								39	14
Area (metal5)									34
Area (no well)	145								
Fringe (substrate)	241	207		--	59	54	41	24	
Fringe (poly)				62	38	29	23	20	17
Fringe (metal1)					55	34		22	19
Fringe (metal2)						50	35	27	22
Fringe (metal3)							52	35	27

[MOSIS website]



```
void print_rlc_line(int *counterstart,int numbero
                    int *pmysecondinductorcounter
                    int *ppointintoconnectarray)
{

  int i,counter1=0,counter2=0,counter3=0;
  int temp;

  for (i=1;i < numberofrcelem;i++){

    if (i==1)
      counter1=*counterstart;
    else
      counter1=counter3;


    counter2=counter1 + 1;
    counter3=counter2 + 1;

    printf("\nR%d %d %d %lg\n",*pmysecondresistor
    *pmysecondresistorcounter = *pmysecondresisto
```

HSPICE Code Generation Tool

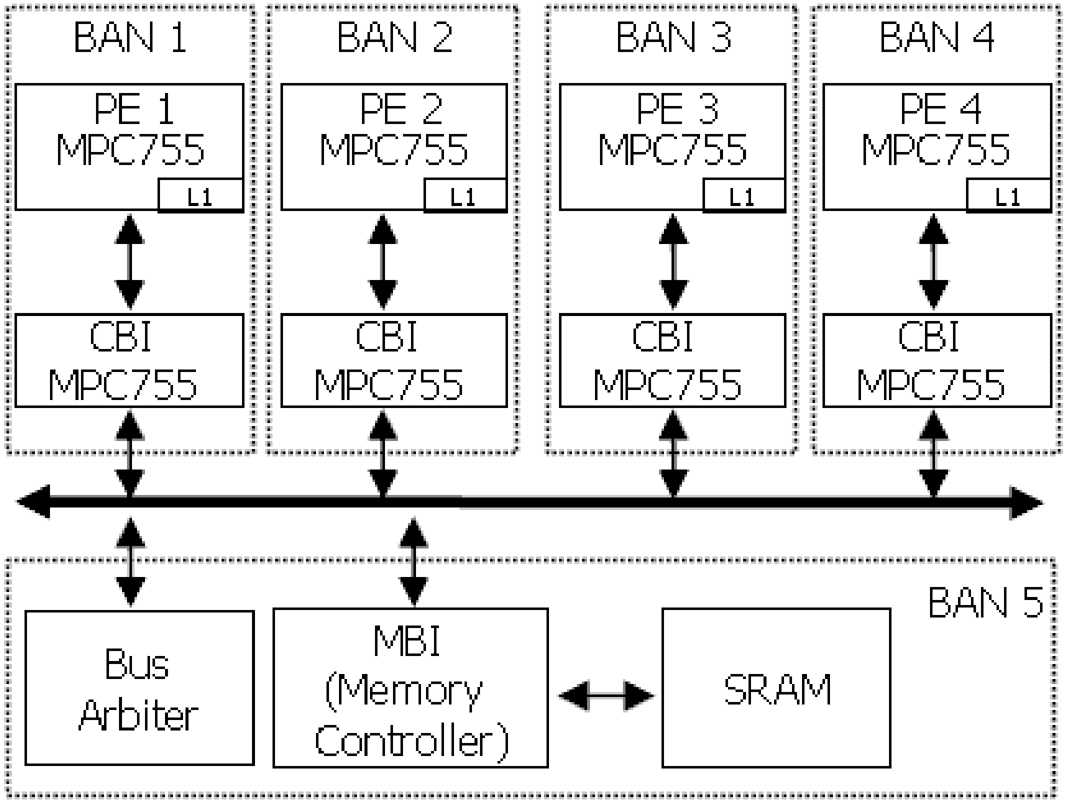


HSPICE simulator

Interconnect Delay Calculation for Each Bus Segment

A Bus System Example:

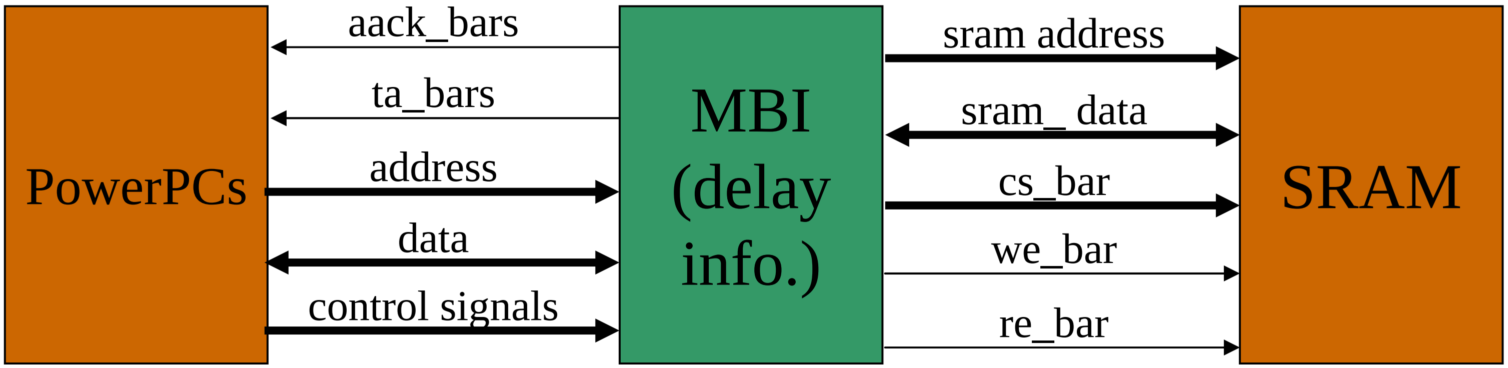
General Global Bus Architecture (GGBA)



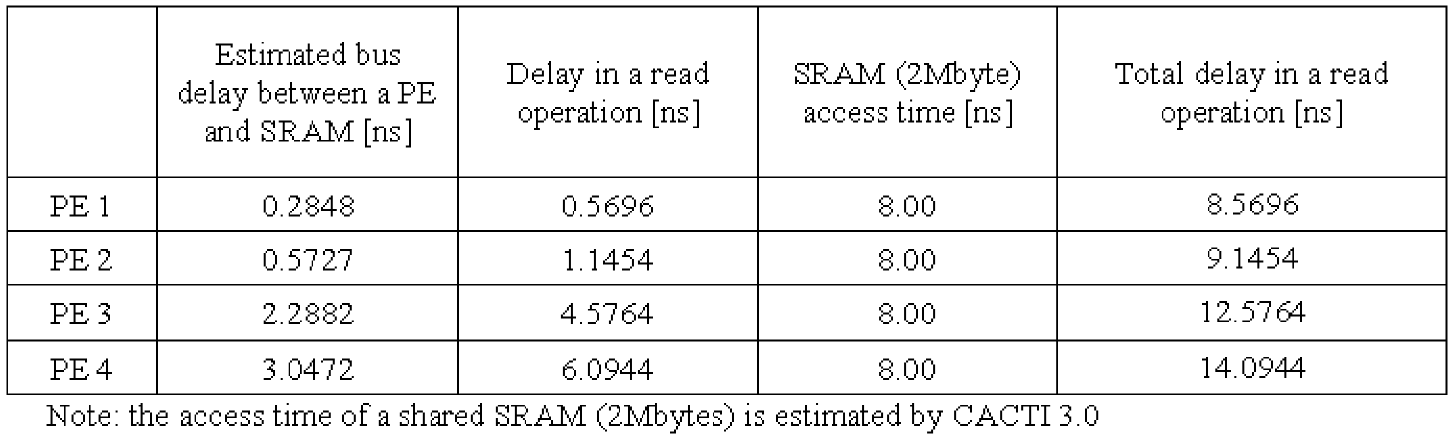
Note BAN: Bus Access Node, PE: Processing Element, CBI: CPU Bus Interface
MBI: Memory Bus Interface

Memory Bus Interface (MBI) Module Generation 1

- One of effects of interconnect delay insertion in an SoC: memory access cycle
- Memory controller to adapt delay clocks due to interconnect delay



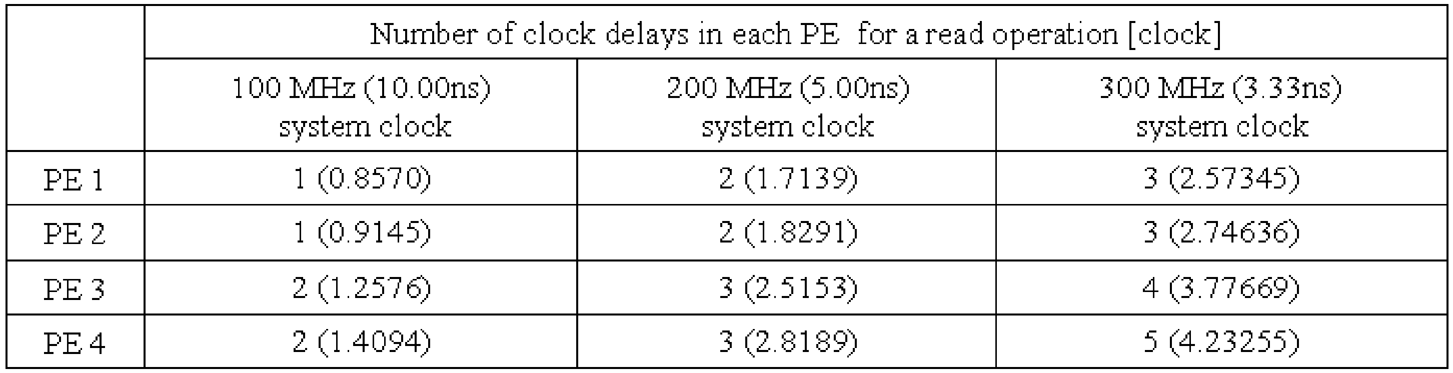
Memory Bus Interface (MBI) Module Generation 2



	Estimated bus delay between a PE and SRAM [ns]	Delay in a read operation [ns]	SRAM (2Mbyte) access time [ns]	Total delay in a read operation [ns]
PE 1	0.2848	0.5696	8.00	8.5696
PE 2	0.5727	1.1454	8.00	9.1454
PE 3	2.2882	4.5764	8.00	12.5764
PE 4	3.0472	6.0944	8.00	14.0944

Note: the access time of a shared SRAM (2Mbytes) is estimated by CACTI 3.0

(a) Estimated total delay of paths between each PE and a shared memory

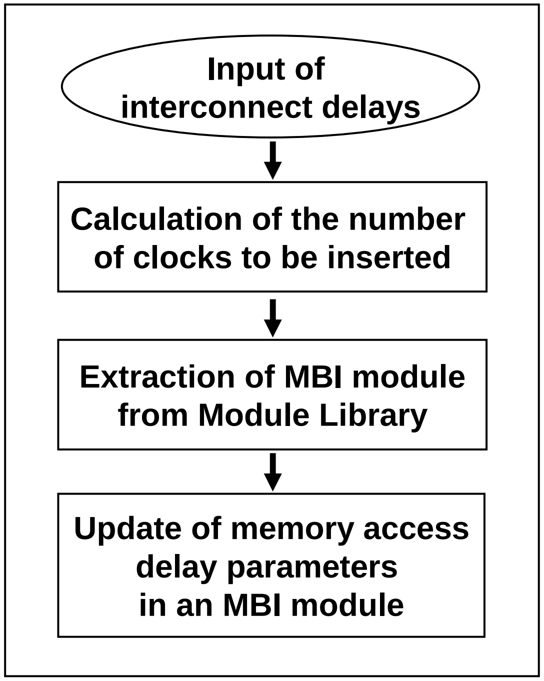


	Number of clock delays in each PE for a read operation [clock]		
	100 MHz (10.00ns) system clock	200 MHz (5.00ns) system clock	300 MHz (3.33ns) system clock
PE 1	1 (0.8570)	2 (1.7139)	3 (2.57345)
PE 2	1 (0.9145)	2 (1.8291)	3 (2.74636)
PE 3	2 (1.2576)	3 (2.5153)	4 (3.77669)
PE 4	2 (1.4094)	3 (2.8189)	5 (4.23255)

(b) Number of clock delays in data paths

MBI and Bus System Generation

- Memory Bus Interface (MBI) module generation



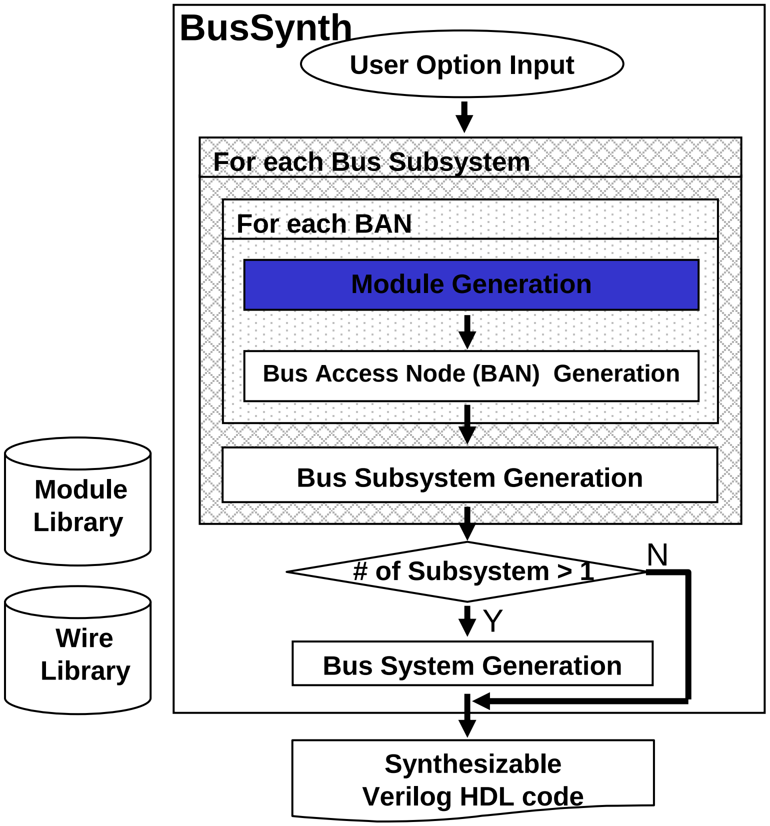
Input of interconnect delays

Calculation of the number of clocks to be inserted

Extraction of MBI module from Module Library

Update of memory access delay parameters in an MBI module

(a) Sequence of MBI Generation



BusSynth

User Option Input

For each Bus Subsystem

For each BAN

Module Generation

Bus Access Node (BAN) Generation

Bus Subsystem Generation

Module Library

Wire Library

# of Subsystem > 1

N

Y

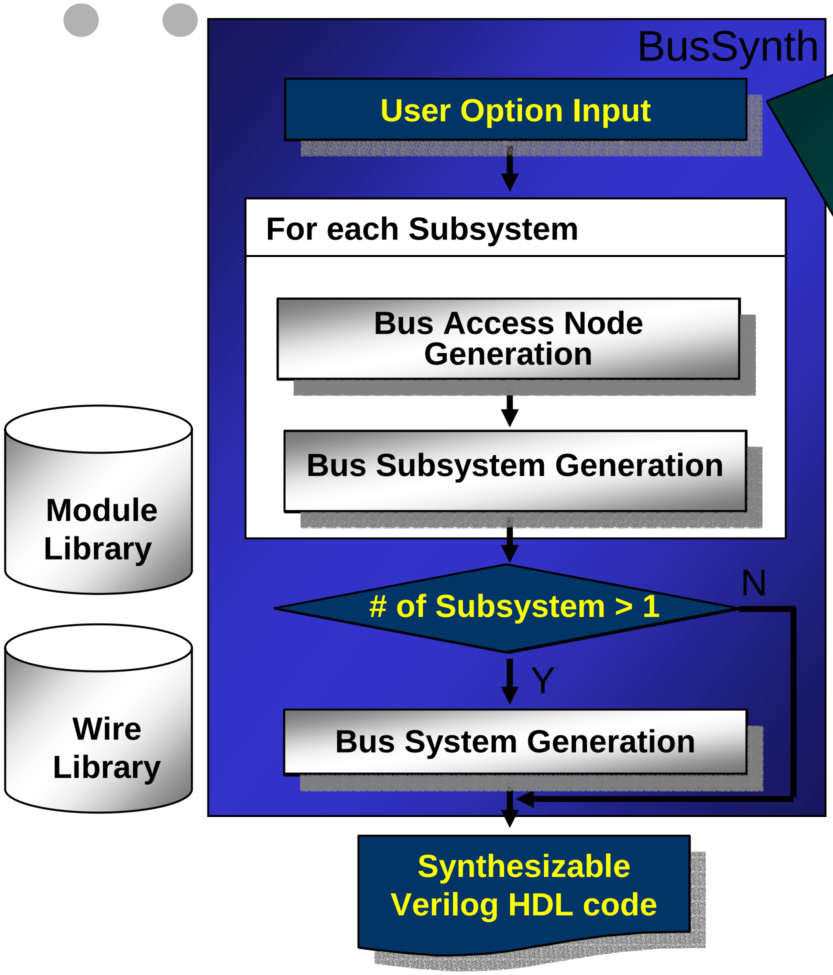
Bus System Generation

Synthesizable Verilog HDL code

(b) Bus System Generation*

Reference*: K. Ryu and V. Mooney, “Automated Bus Generation for Multiprocessor SoC Design,” *Design, Automation and Test in Europe* (DATE'03), pp. 282-287, March 2003.

A Bus System Generation Example

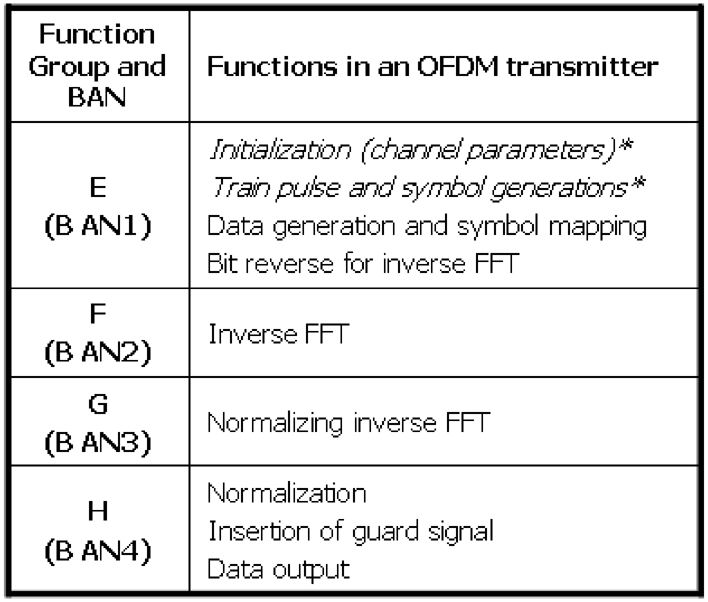


User Input List

```
        //  Skipped
         .up_dataout(dataout_up_2[FIFO_D_WIDTH-1:0]),
         .up_gen_int(gen_int_up_2),
         .up_isr0_ctlhi(isr0_ctlhi_up_2),
         .up_isr0_ctllo(isr0_ctllo_up_2),
         .dn_datain(datain_up_3[FIFO_D_WIDTH-1:0]),
         .reb_dn(reb_up_3),
         .web_dn(web_up_3),
         .fifo_area_dn(fifo_area_up_3)
  );
endmodule

module BusSystem(sysrstb, sysclk);
  input sysrstb;
  input sysclk;
  // Skipped

  SubSys_GGBA SubSystem(
        .sysrstb(sysrstb),
        .sysclk(sysclk)
        // Skipped
   );

endmodule
```

Application Example

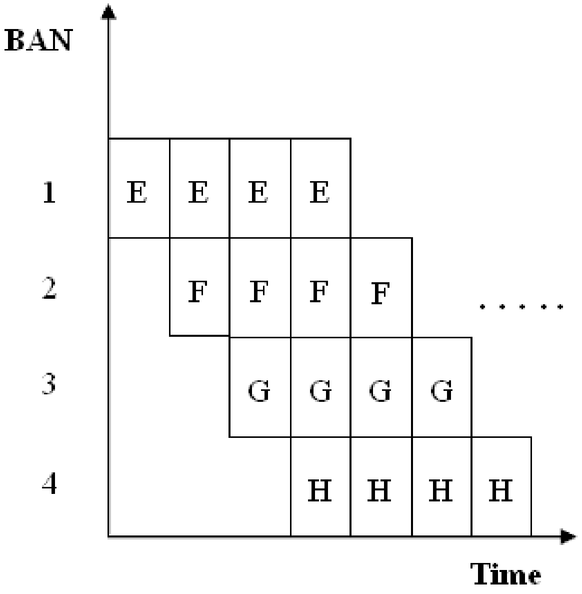
- Orthogonal Frequency Division Multiplexing (OFDM) Transmitter, a wireless algorithm
- Function assignment and their processing



Function Group and BAN	Functions in an OFDM transmitter
E (B AN1)	_Initialization (channel parameters)*_ _Train pulse and symbol generations*_ Data generation and symbol mapping Bit reverse for inverse FFT
F (B AN2)	Inverse FFT
G (B AN3)	Normalizing inverse FFT
H (B AN4)	Normalization Insertion of guard signal Data output

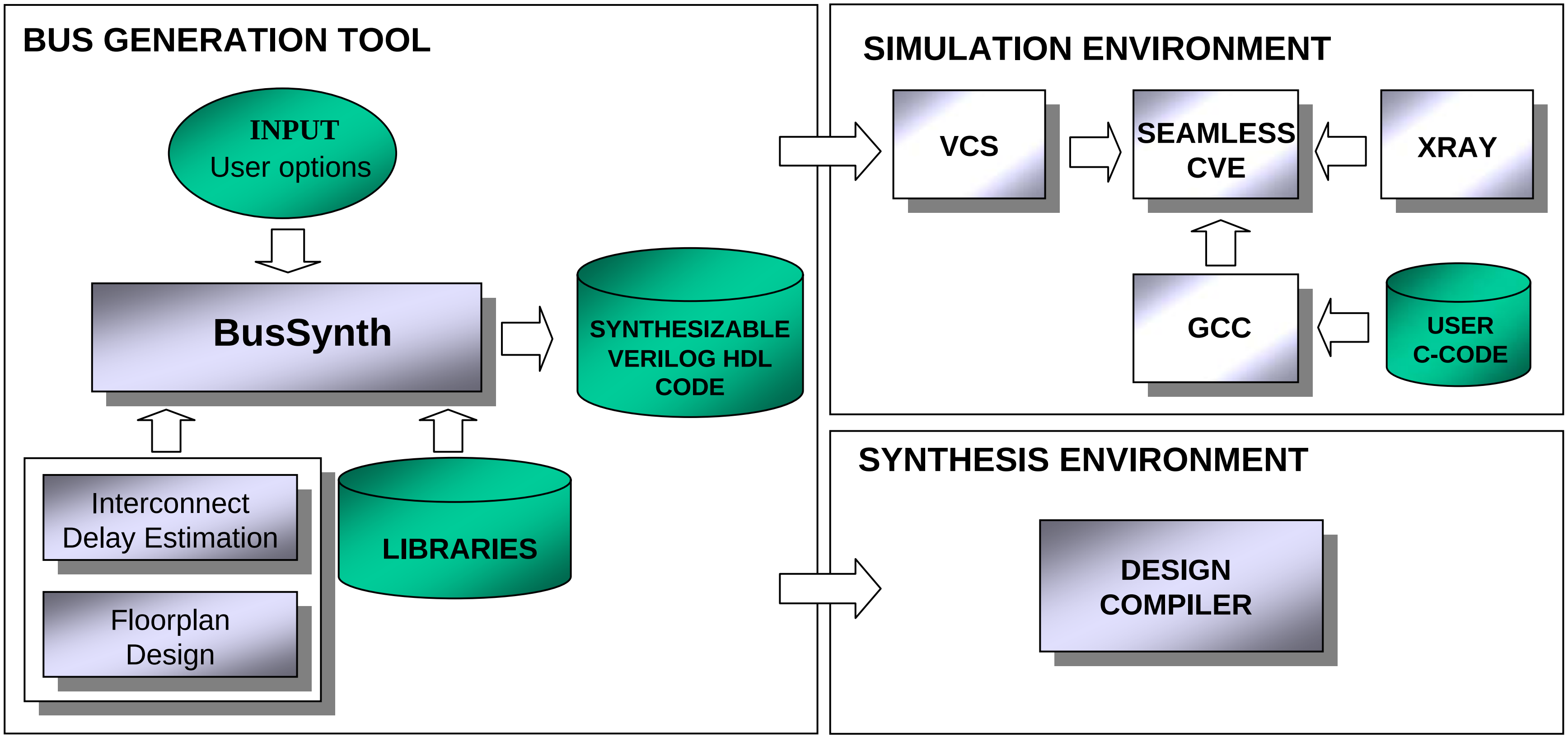
*Italicized functions are executed once at first

(a) Function group in OFDM transmitter



(b) Pipelined parallel processing

Experimental Setup



Note: VCS and Design Compiler from Synopsys, Seamless CVE and Xray from Mentor Graphics and GCC from GNU

Three Configurations of GGBA for Performance Comparison

- **GGBA I - (NO WIRE MODEL)**

  GGBA I is a GGBA system with no regard to interconnect delay on the bus

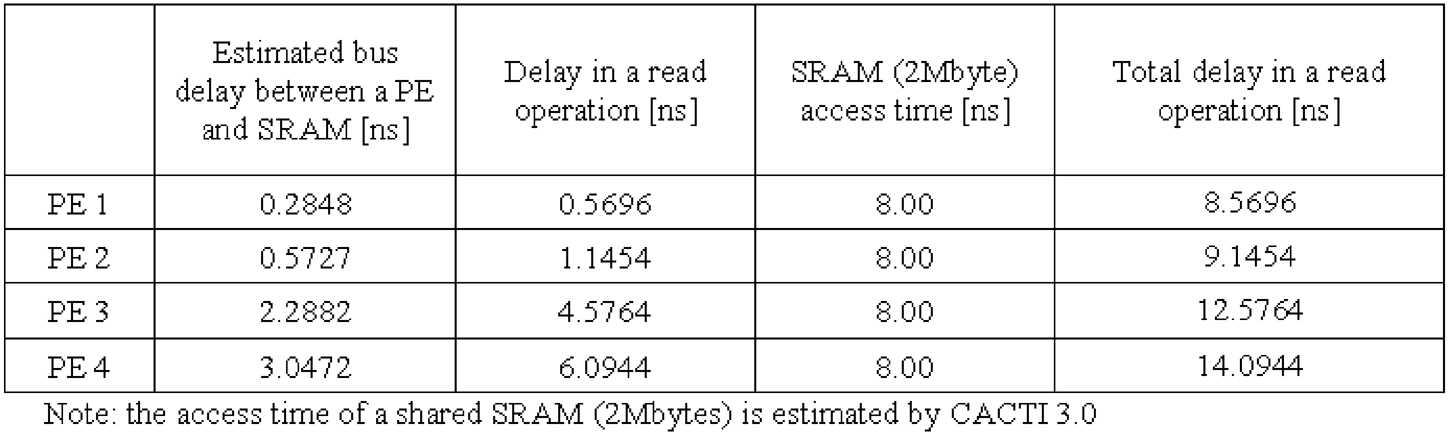
- **GGBA II - (ACCURATE WIRE MODEL)**

  GGBA II is a GGBA system that works with different estimated interconnect delays on the shared bus

- **GGBA III - (WORST-CASE WIRE MODEL)**

  GGBA III is a GGBA system that operates with a maximum estimated delay on all connections between PEs and a shared memory

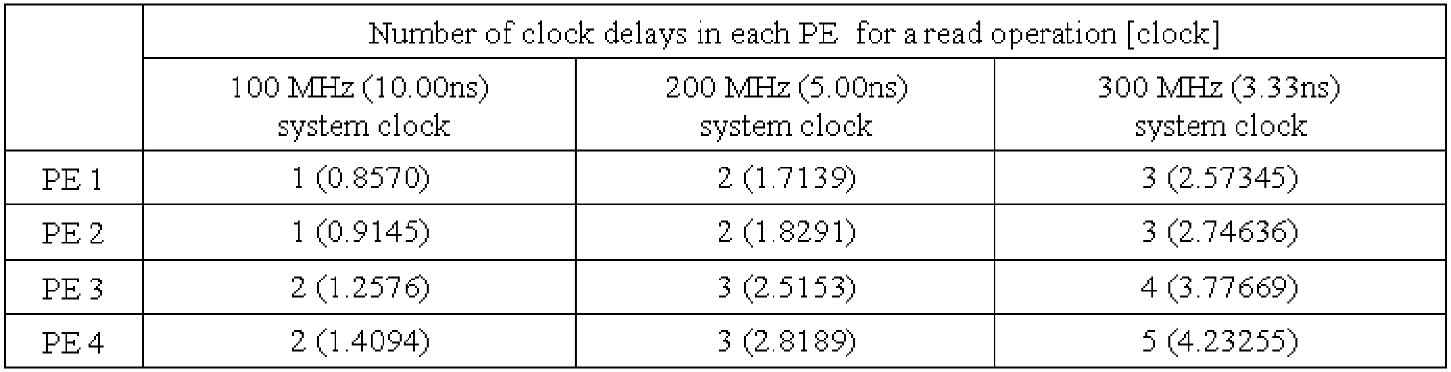
Memory Bus Interface (MBI) Module Generation 2



	Estimated bus delay between a PE and SRAM [ns]	Delay in a read operation [ns]	SRAM (2Mbyte) access time [ns]	Total delay in a read operation [ns]
PE 1	0.2848	0.5696	8.00	8.5696
PE 2	0.5727	1.1454	8.00	9.1454
PE 3	2.2882	4.5764	8.00	12.5764
PE 4	3.0472	6.0944	8.00	14.0944

Note: the access time of a shared SRAM (2Mbytes) is estimated by CACTI 3.0

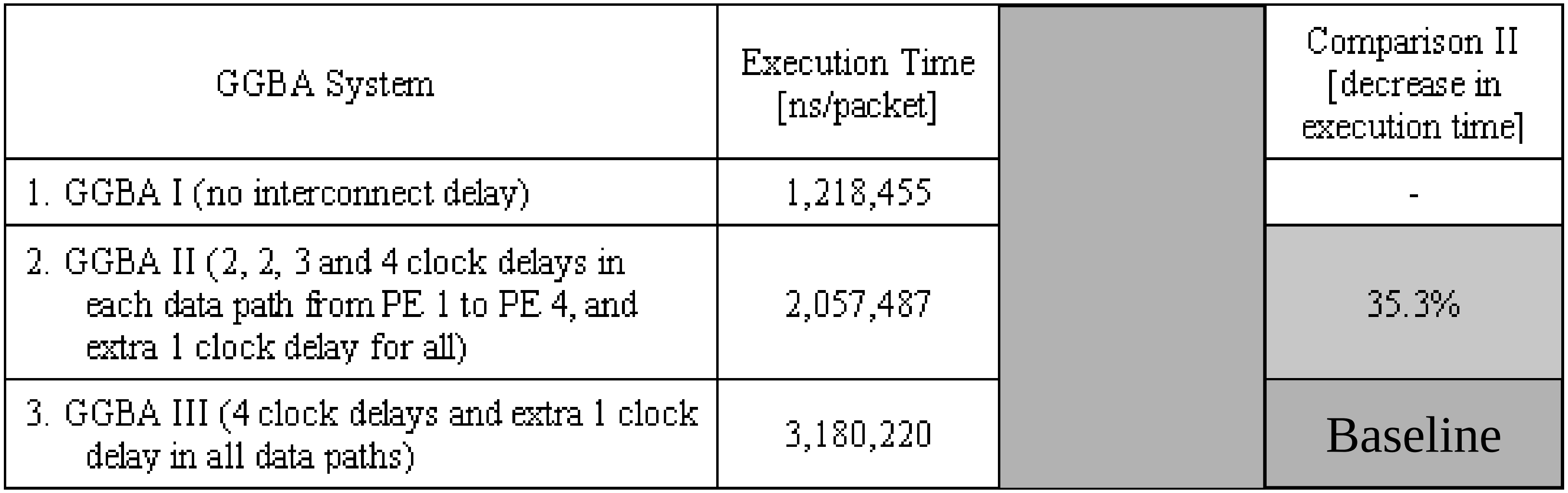
(a) Estimated total delay of paths between each PE and a shared memory



	Number of clock delays in each PE for a read operation [clock]		
	100 MHz (10.00ns) system clock	200 MHz (5.00ns) system clock	300 MHz (3.33ns) system clock
PE 1	1 (0.8570)	2 (1.7139)	3 (2.57345)
PE 2	1 (0.9145)	2 (1.8291)	3 (2.74636)
PE 3	2 (1.2576)	3 (2.5153)	4 (3.77669)
PE 4	2 (1.4094)	3 (2.8189)	5 (4.23255)

(b) Number of clock delays in data paths

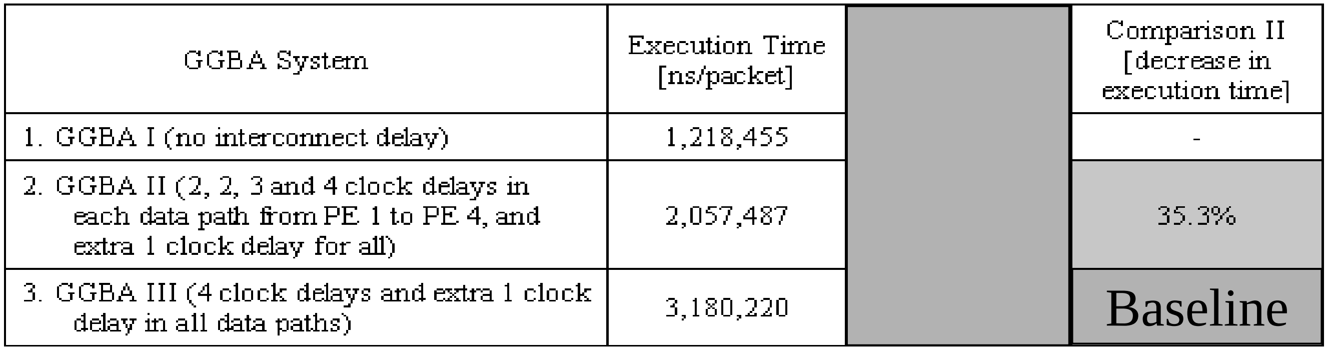
Comparison Results



GGBA System	Execution Time [ns/packet]		Comparison II [decrease in execution time]
1. GGBA I (no interconnect delay)	1,218,455		-
2. GGBA II (2, 2, 3 and 4 clock delays in each data path from PE 1 to PE 4, and extra 1 clock delay for all)	2,057,487		35.3%
3. GGBA III (4 clock delays and extra 1 clock delay in all data paths)	3,180,220		Baseline

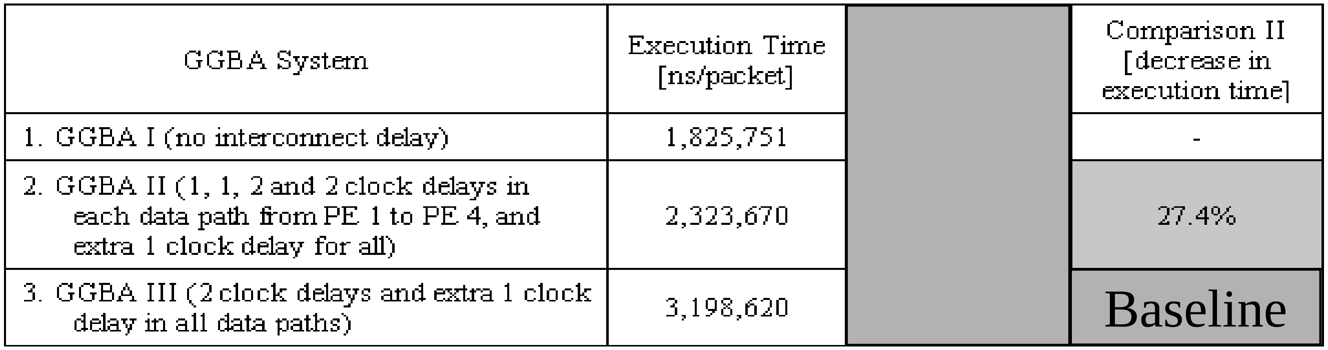
(a) 300 MHz Bus Clock

Comparison Results



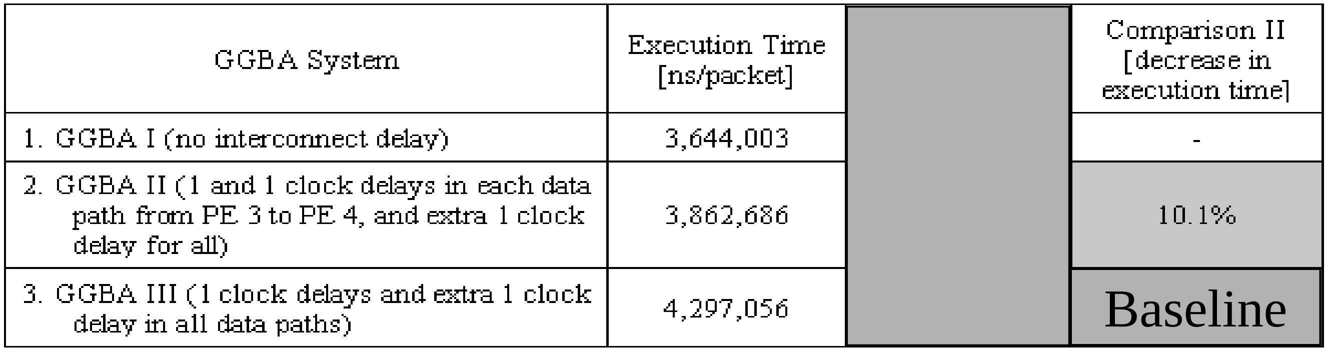
GGBA System	Execution Time [ns/packet]		Comparison II [decrease in execution time]
1. GGBA I (no interconnect delay)	1,218,455		-
2. GGBA II (2, 2, 3 and 4 clock delays in each data path from PE 1 to PE 4, and extra 1 clock delay for all)	2,057,487		35.3%
3. GGBA III (4 clock delays and extra 1 clock delay in all data paths)	3,180,220		Baseline

(a) 300 MHz Bus Clock



GGBA System	Execution Time [ns/packet]		Comparison II [decrease in execution time]
1. GGBA I (no interconnect delay)	1,825,751		-
2. GGBA II (1, 1, 2 and 2 clock delays in each data path from PE 1 to PE 4, and extra 1 clock delay for all)	2,323,670		27.4%
3. GGBA III (2 clock delays and extra 1 clock delay in all data paths)	3,198,620		Baseline

(b) 200 MHz Bus Clock



GGBA System	Execution Time [ns/packet]		Comparison II [decrease in execution time]
1. GGBA I (no interconnect delay)	3,644,003		-
2. GGBA II (1 and 1 clock delays in each data path from PE 3 to PE 4, and extra 1 clock delay for all)	3,862,686		10.1%
3. GGBA III (1 clock delays and extra 1 clock delay in all data paths)	4,297,056		Baseline

(c) 100 MHz Bus Clock

Conclusion

- Interconnect delay is a major concern as feature size is scaled down
- Interconnect delay estimation from floorplan
- Memory Bus Interface (MBI) module and Bus System generation
- Performance improvement due to interconnect delay aware design
- In an OFDM transmitter example, 35.3% reduction in execution time against GGBA III

Any Questions ?