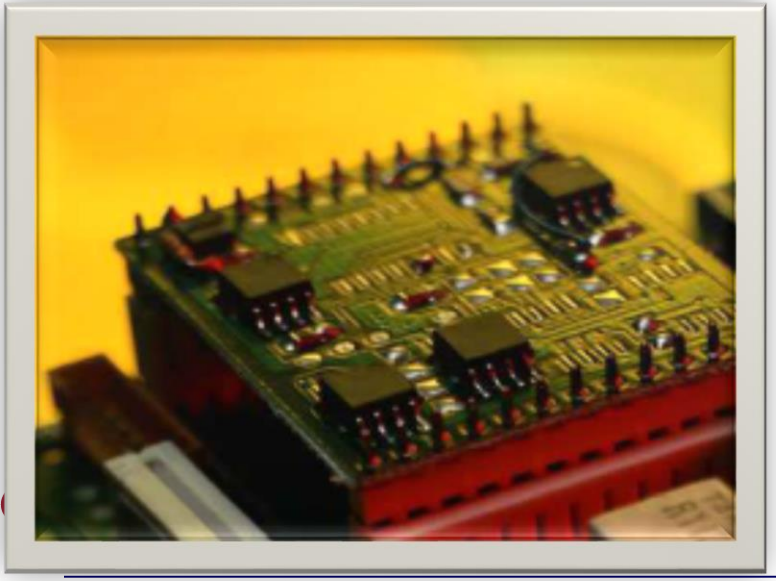
THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

In Situ Programmable SoC Design Space Exploration



Supervisor: ***Dr. Benjamin Carrion Schafer***

Student: ***Siyuan XU***

Major: Electronic and Information Engineering

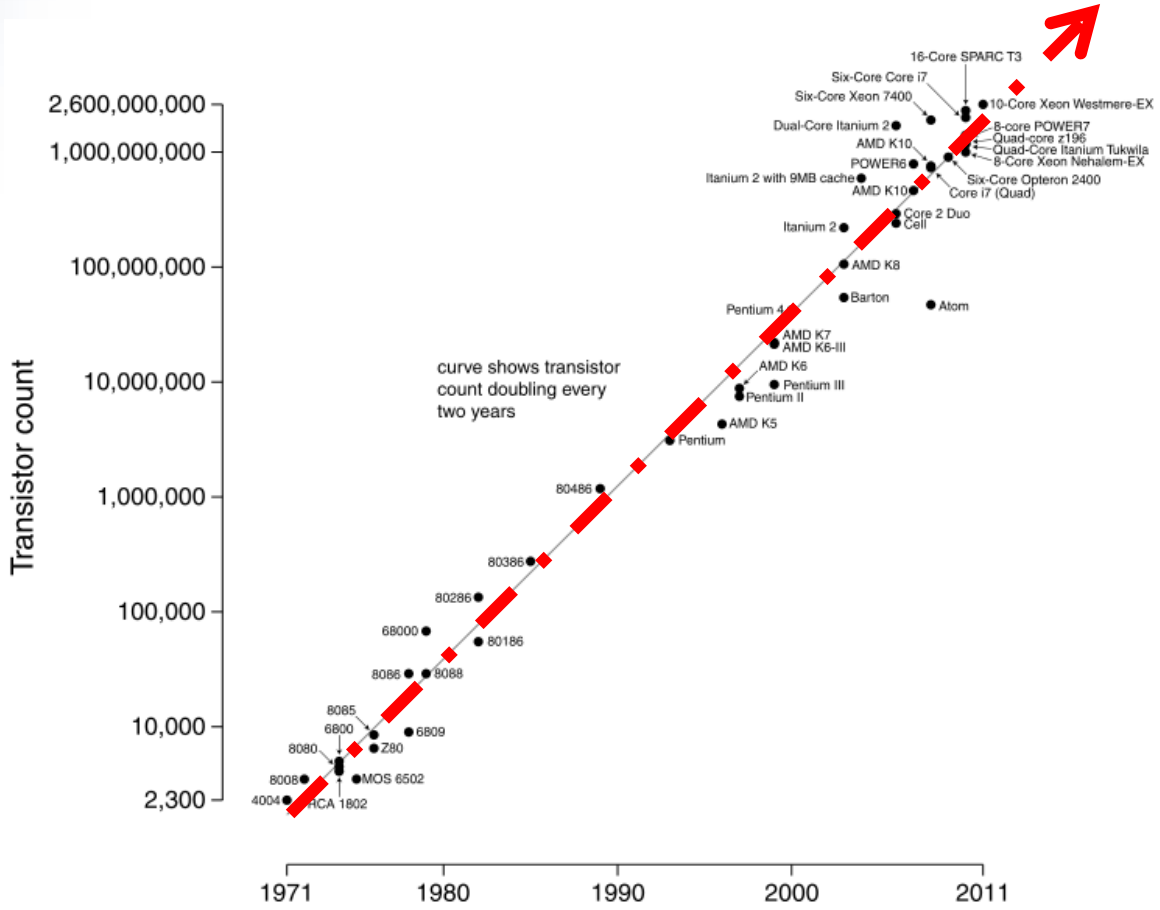
Thesis Architecture

1. Introduction
2. High Level Synthesis
3. Configurable System-on-Chip
4. AS2CBench
5. In Situ DSE
6. Conclusions and Future Work

Introduction

◆ Gordon Moore: co-founder of Intel

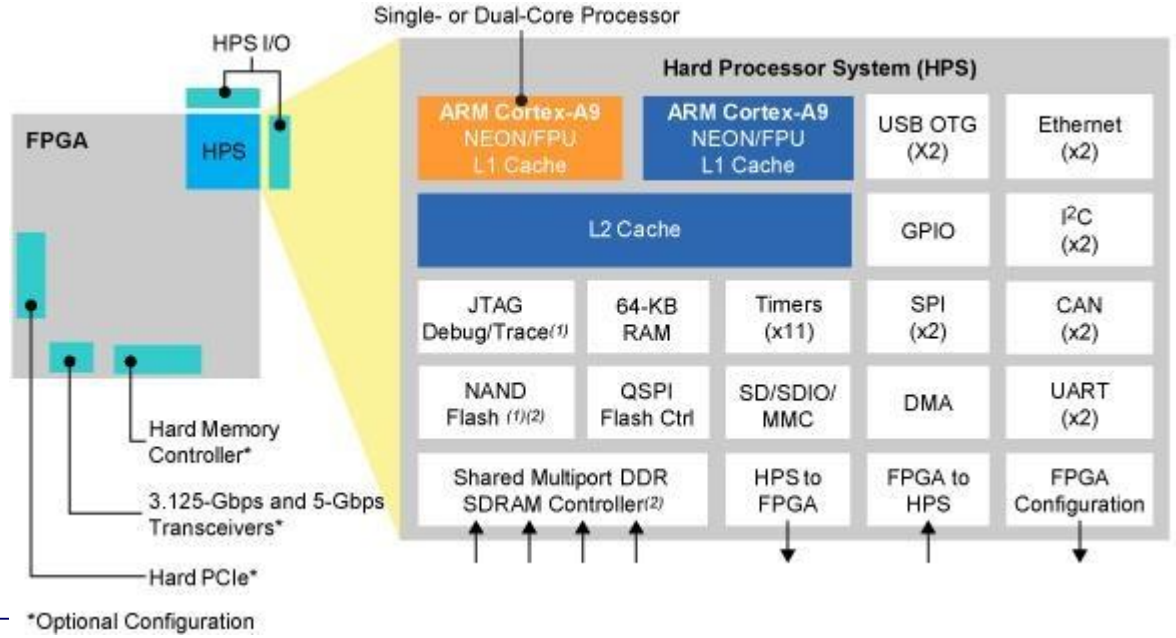
Predicted that number of transistors per chip would grow exponentially (double every 18-24 months).



Introduction

- Most circuits are now heterogeneous Multiprocessor Systems-on-Chips (MPSoCs)

1. Embedded microprocessors.
2. Memory controllers.
3. Memories .
4. Dedicated hardware accelerators.



Introduction

- Due to the pressure to tape-out these chips at shorter design cycles, companies often rely on third party Intellectual Properties (3PIPs) to meet their tight schedules.
- High-Level Synthesis (HLS) to increase their design productivity.

Vendor	Tool Name	Supported Languages
Cadence (Forte)	Cynthesizer	SystemC
Cadence	C-to-Silicon	C, C++, SystemC
Calypto	CatapultC	C++, SystemC
NEC	CyberWorkBench	C, SystemC
Xilinx	Vivado HLS	C, C++, SystemC

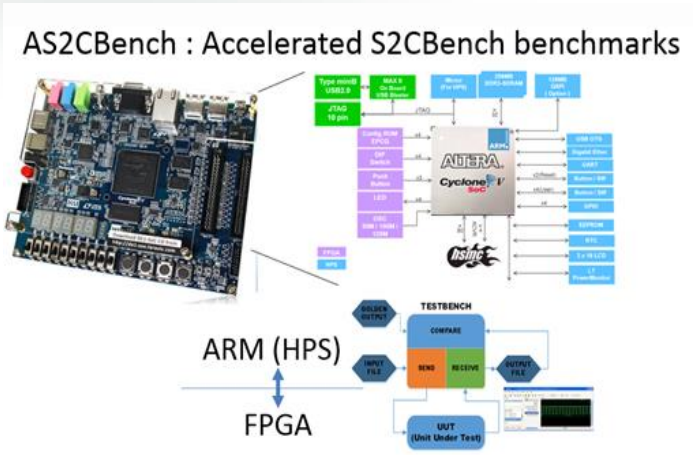
At the same time SystemC (C++ class for HW design) has emerged as a common language for these HLS tools

In 2013, S2CBench benchmark suite Release

Contribution --1

S2CBench benchmark suite

Test cases to perform HLS Design Space Exploration (DSE) experiments at the department.



Deals with HW/SW co-design

In particular system-level DSE.(CSoC)

No available benchmarks are available

AS2CBench benchmark suite

Benchmark suite for HW/SW co-design, especially on CSoCs

Contribution --2

A Cycle-Accurate Model

PC

1 . Slow :
PC executes sequentially

2. Can't be exactly modeled,
Hence the results can slightly differ.

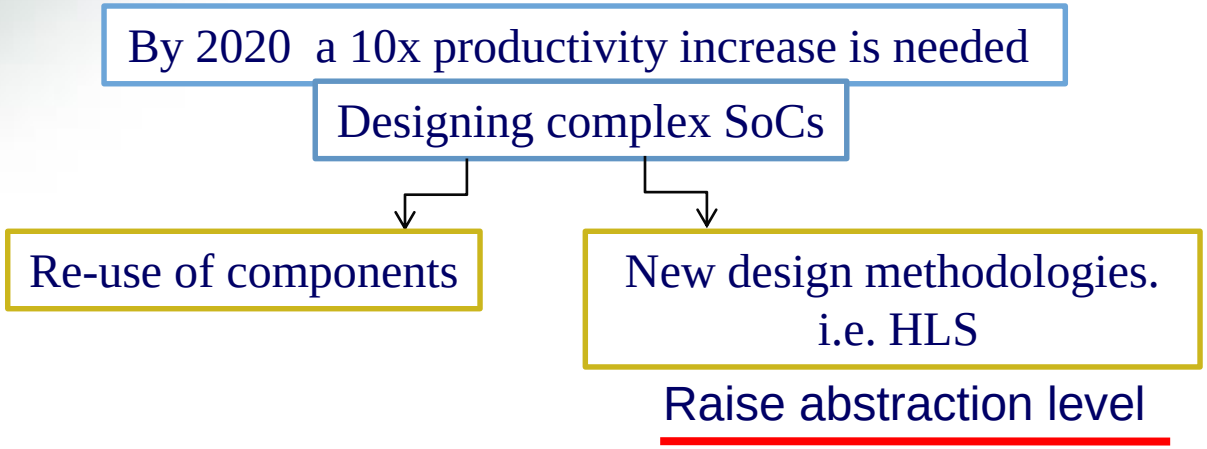
Prototyping on real configurable SoCs,
e.g. Altera's Cyclone V SoC or Xilinx's Zynq FPGA.

It would be interesting to compare the simulation based results with the prototyped ones.

100% Accurate

Actual final platform

International Technology Roadmap for Semiconductors (ITRS)



The following will introduce how HLS is applied in HW/SW design on heterogeneous system.

High Level Synthesis Introduction

Behavioral Model
(ANSI-C/C++/SystemC)

High Level Synthesis

1. Allocation
2. Scheduling
3. Binding

Register Transfer Level (RTL)
(Verilog/VHDL)

Logic Synthesis

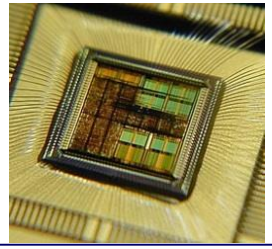
Gate Level Netlist

Physical Synthesis
(Place and Route)

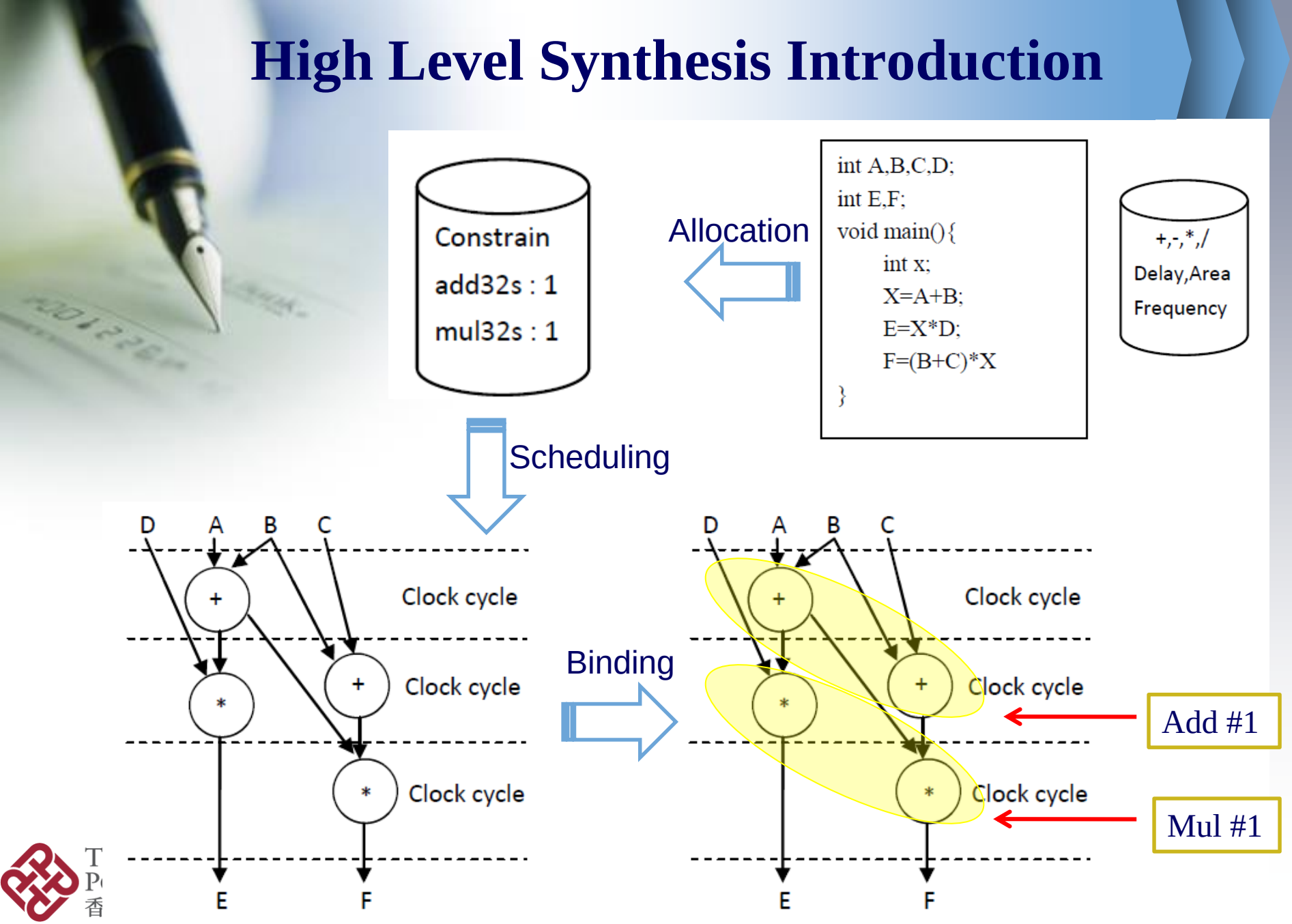
GDSII/Bitstream
(ASIC/FPGA)

Why HLS?

1. Reduces the complexity of hardware design
2. Less number of lines of code are required.
3. Less bug..
4. Easy to verify

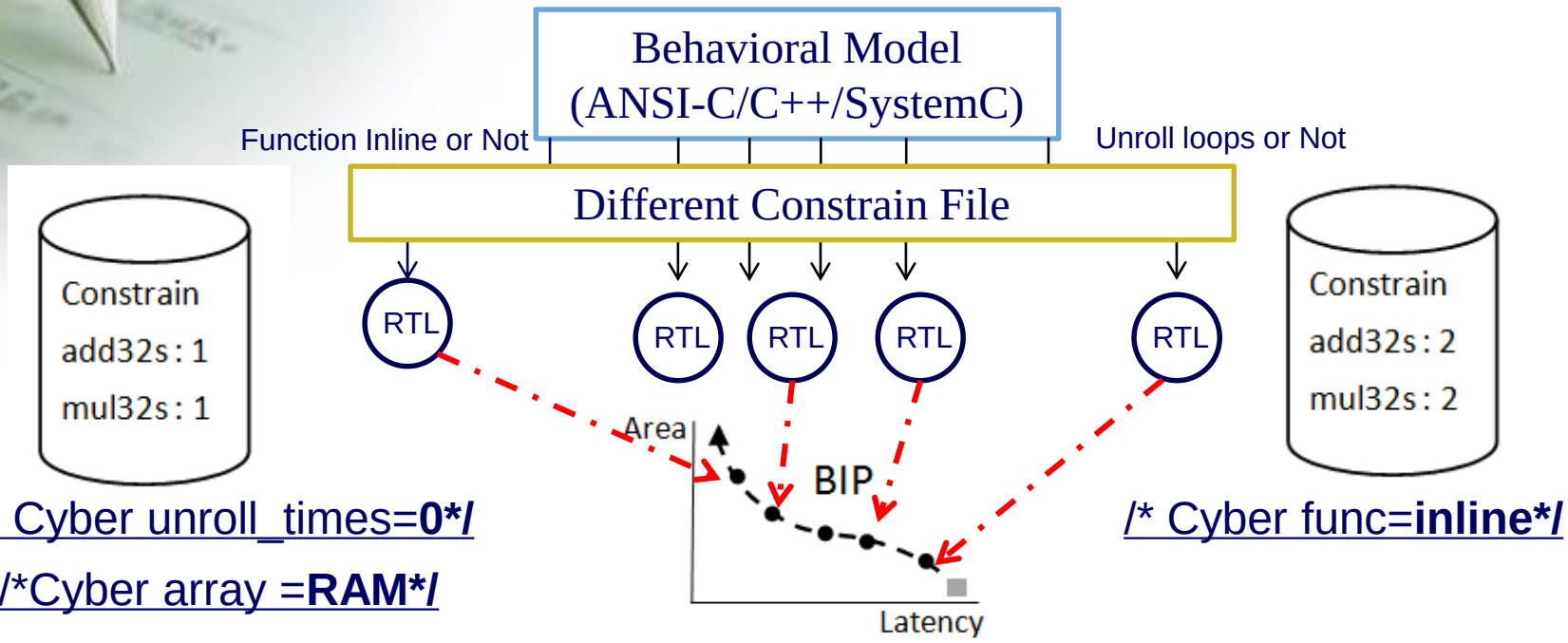


High Level Synthesis Introduction



High Level Synthesis Introduction

- Area vs. Performance trade-offs
- Without having to modify the original behavioral description.

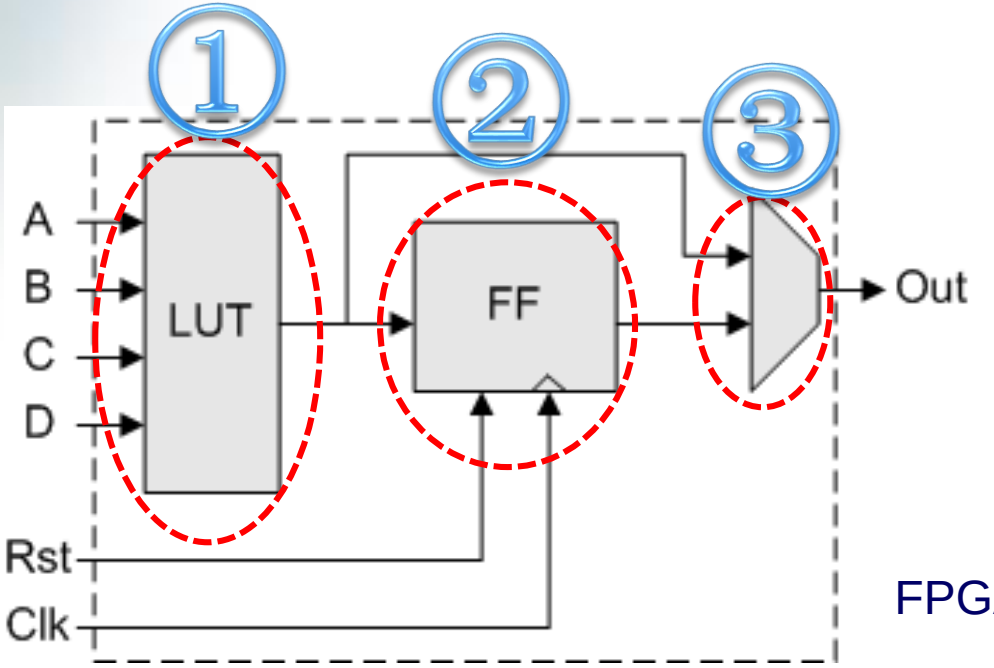


BIP trade-off curves

- This unique features will be leveraged in the thesis.

FPGA Basic Structure

In 1984, Xilinx introduced the modern Look up Table (LUT) based FPGA architecture.

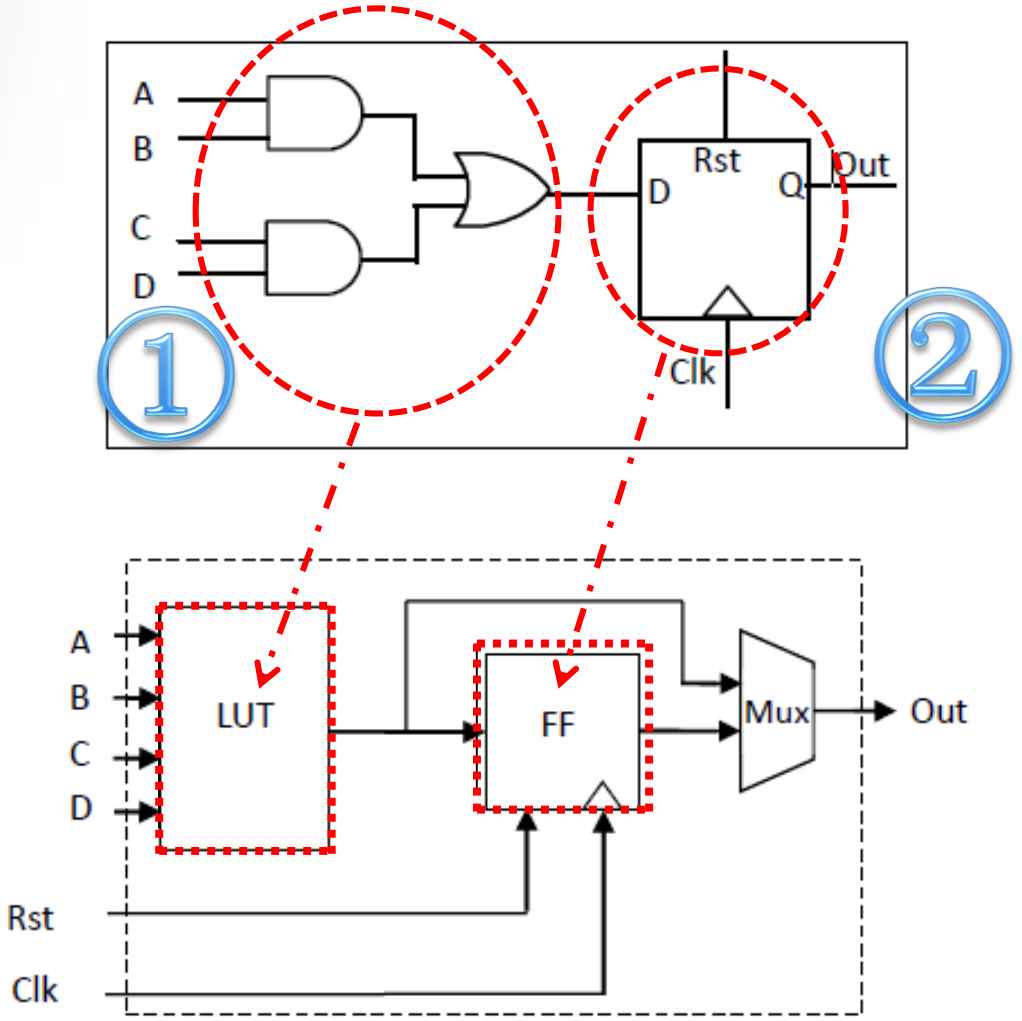


FPGA Basic Structure

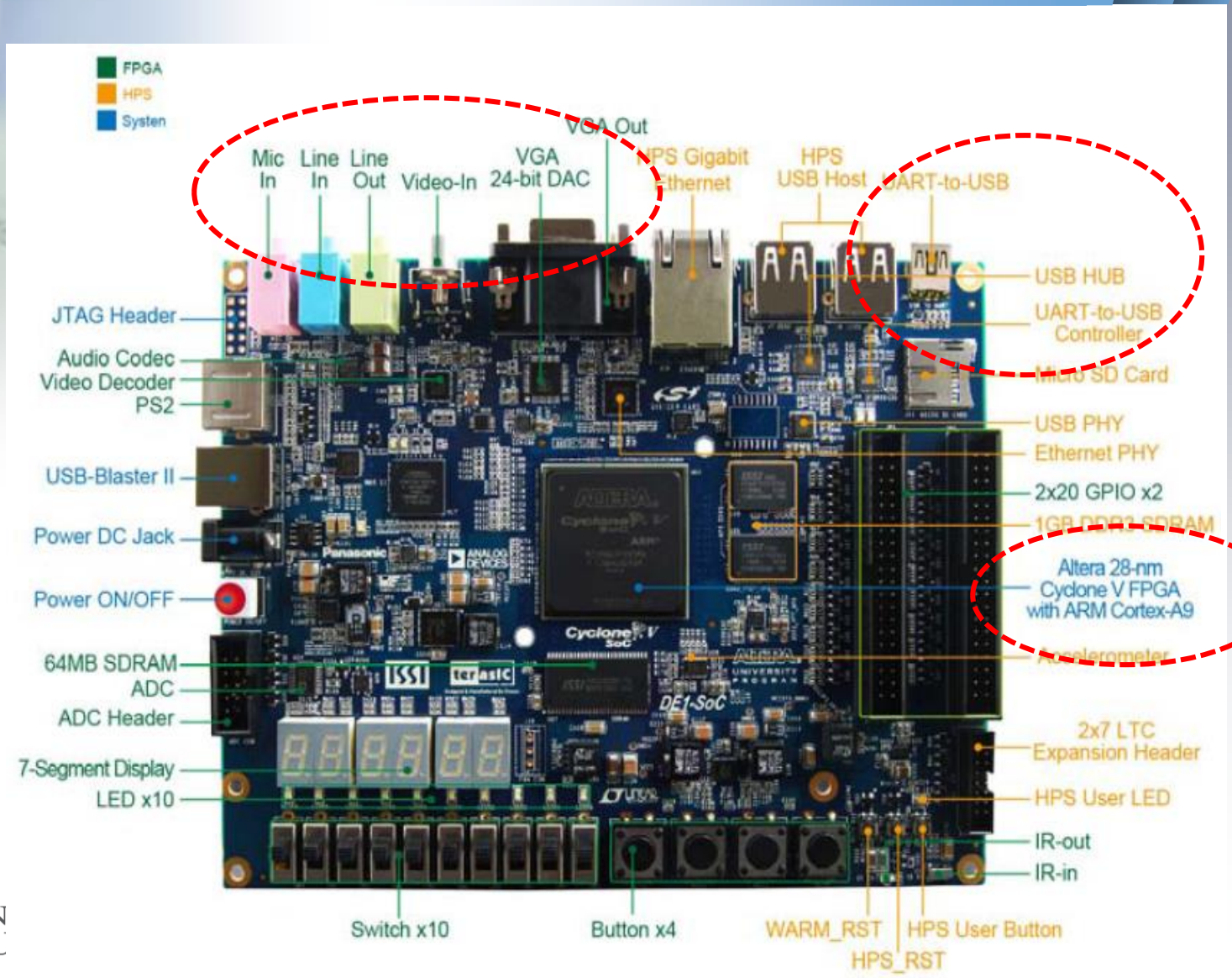
1. 4-input 1-output LUT.
2. Flip-Flop(FF) to register the LUT output.
3. Multiplex(Mux) to use registered output or non-registered.

FPGA Basic Structure

Map the following gate netlist onto a 4-input LUT FPGA



Configurable System-on-Chip



AS2CBench

Table 5.1: AS2Cbench Benchmark Characterization I

Design	SW [lines]	HW [lines] SystemC	HW [lines] Verilog	Latency
sobel	529	364	757	5
fir	428	303	1,255	12
qsort	418	308	1,151	12
adpcm	418	337	1,507	7
kasumi	456	554	65,423	36
snow3G	428	623	5,883	4
ann(2layers)	1,579	472	12,832	6
ann(4layers)	1,584	471	27,141	9
idct	492	1,614	39,704	258
aes	490	645	31,699	105
md5c	452	577	19,996	72
decim	456	549	6,142	23
interp	429	341	1,950	7
disparity	788	580	20,679	416
avg.	639	553	16,866	337

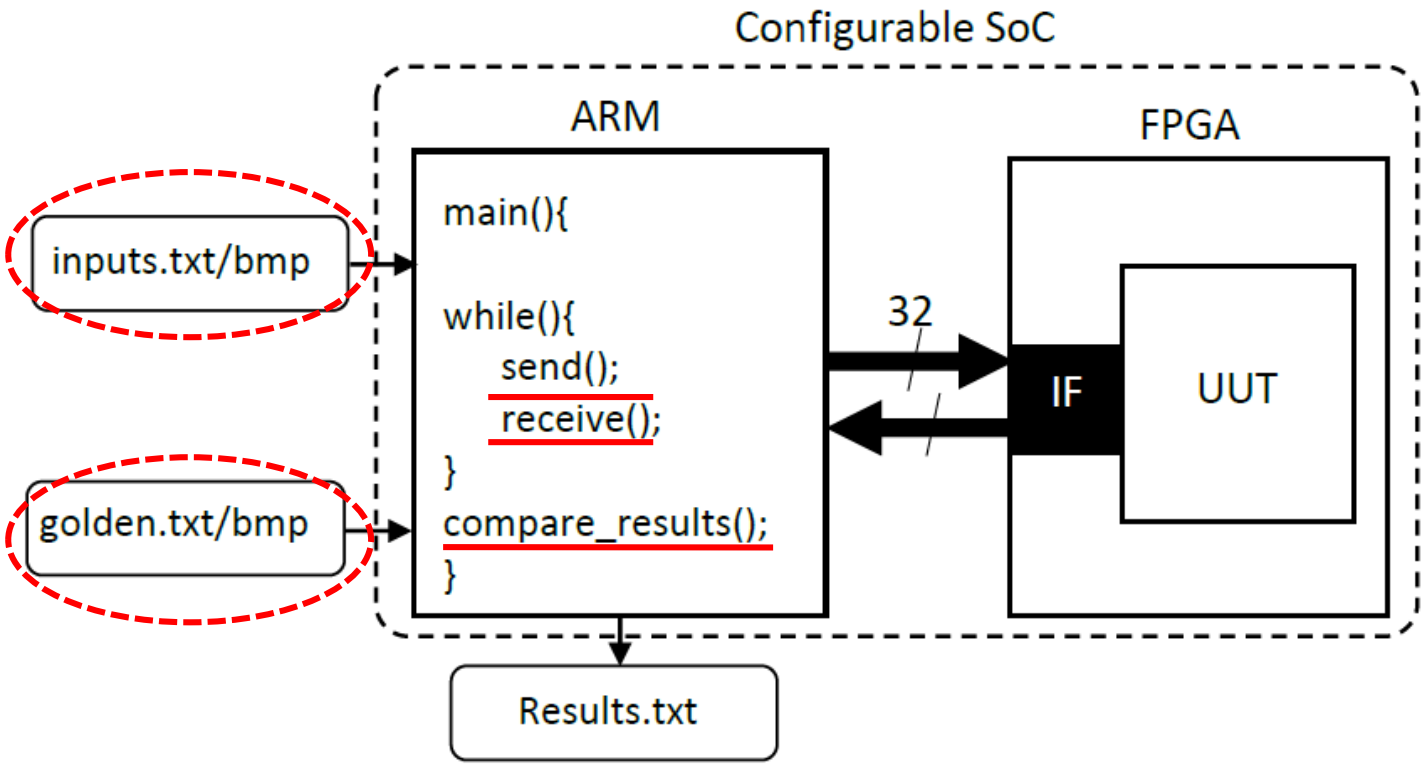
Sobel: Sobel is a 3x3 edge detection filter.
Qsort: Sorts packets of ten numbers in ascending order using a quicksort algorithm.
Kasumi: Kasumi is a block cipher algorithm used in mobile communication systems.
snow3G: Snow3G is a stream cipher which produces a key stream consisting of 32-bit blocks using a 128-bit key.
Interp: This design is a 4-stage interpolation filter
Ann: Artificial Neuronal Networks (ANN)

IDCT: IDCT is a inverse discrete cosine transform
Adpcm: Adaptive Differential Pulse-Code Modulation
Disparity: This program estimates the disparity in a stereoscopic image.

AS2CBench

◆ Accelerated Synthesizable SystemC benchmark suite

The benchmarks are open source and can be downloaded from [2].



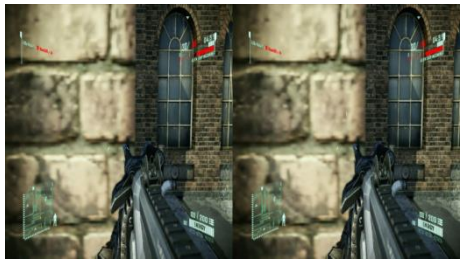
AS2CBench

32bit AXI Bus

Master

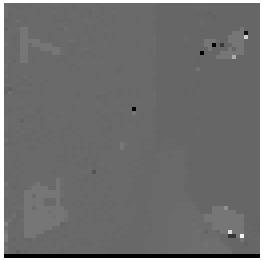
```
for(num=0;num<WDSIZE;num++){
alt_write_word(h2p_lw_UUT_addr,WData[num]);
}
```

Sobel



Disparity

```
while(1){
   outvalid=alt_read_word(h2p_lw_outvalid_addr);
if（outvalid==True）{
    for(num=0;num<RDSIZE;num++){
        alt_write_word(h2p_lw_outvalid_addr,true);
       UUT_out[num]  =alt_read_word(h2p_lw_UUT_addr);
 }
break;  }
  }
```



Slave

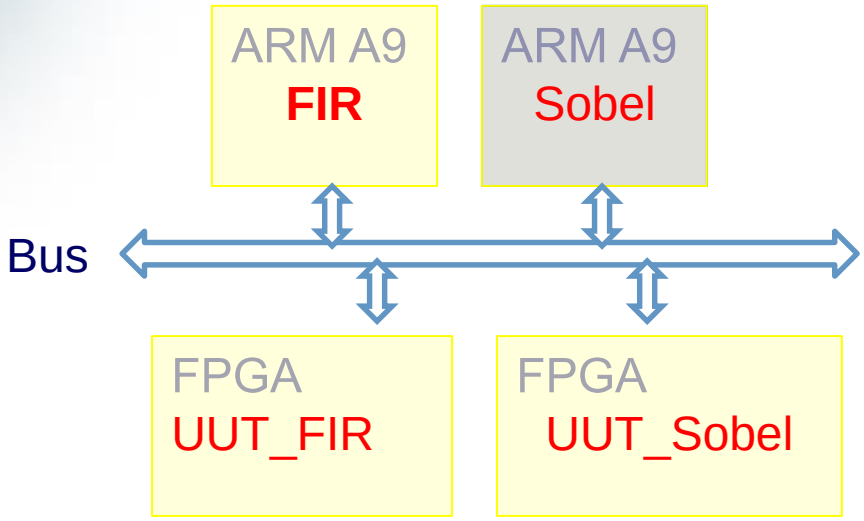
```
while (1) {
input_valid=input_valid_signal.read();/* get status */
if (input_valid == True) {
                WData[num]=UUT_in.read();

                if ((++num) == WDSize)       break;}
}
```

0% … 45% ... …… 100%

```
output_valid_signal.write(True);
while(1){
    output_control =output_control_signal.read();
    if (output_control == True) {
          UUT_out.write(RData[num]);
        if ((++num) == RDSize)                  break;}
}
```

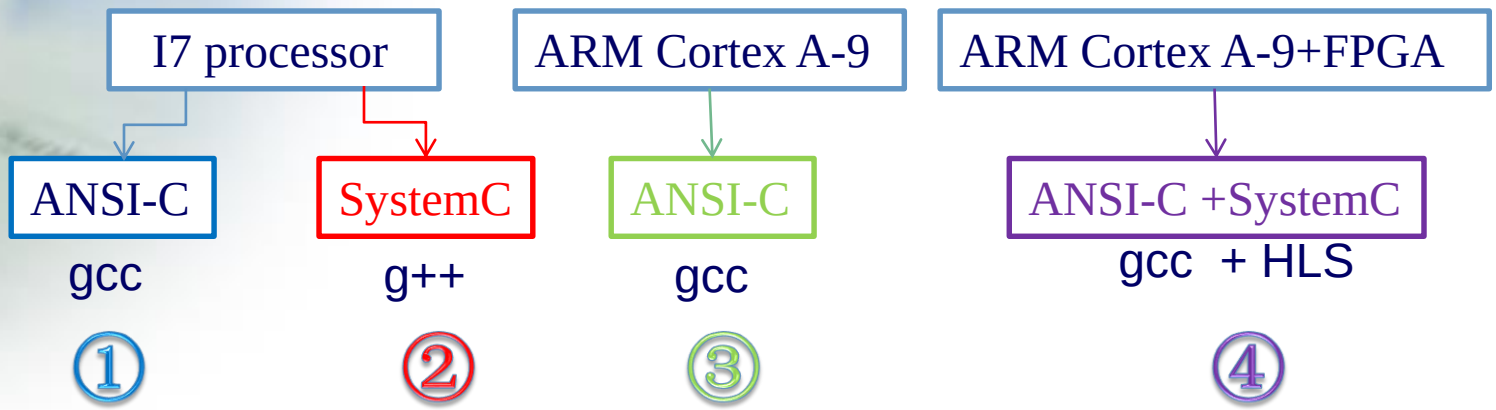
Multi-process Design

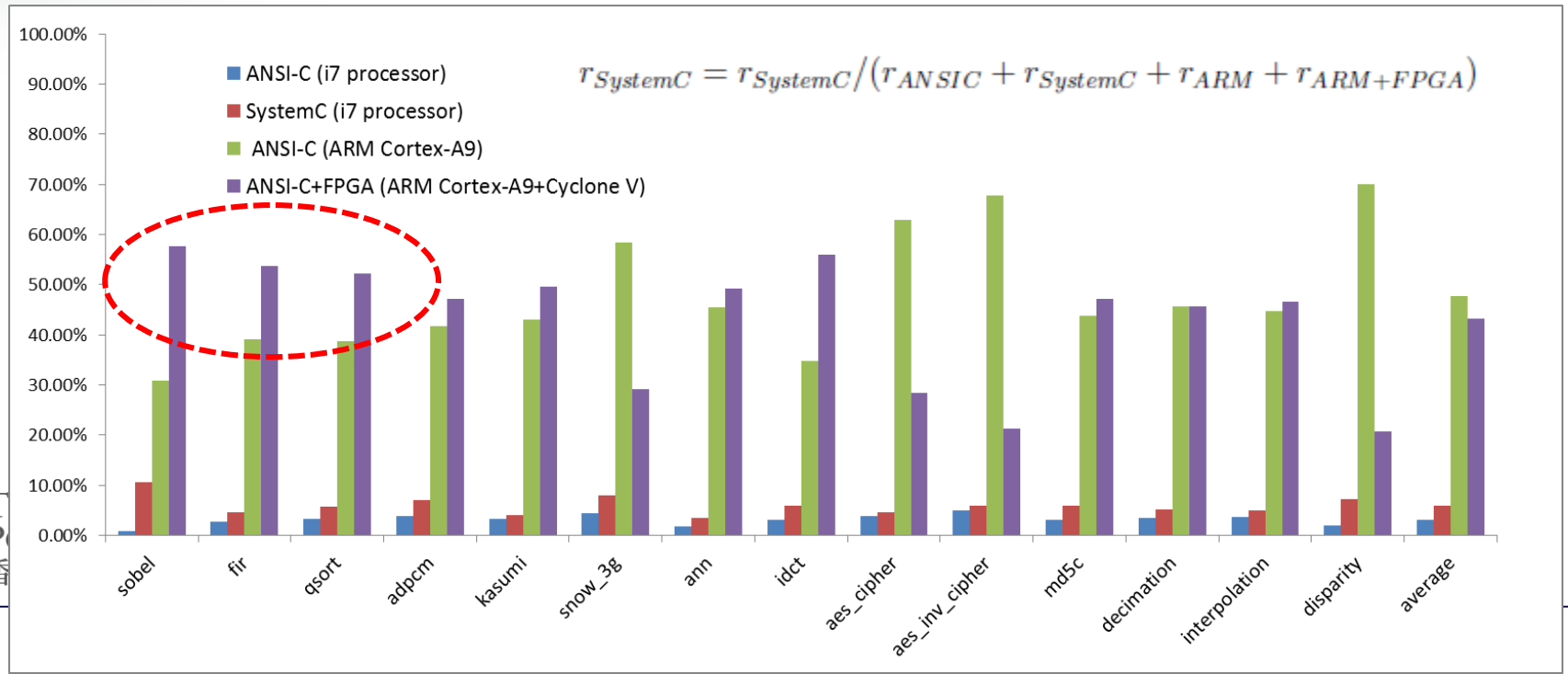


In this work, a design in which two of benchmarks are executed concurrently is also included in the benchmark suite.
In particular the FIR and sobel filter (FIR+sobel).

AS2CBench Result

The first 3 versions are pure SW versions, while the last is the proposed accelerated version

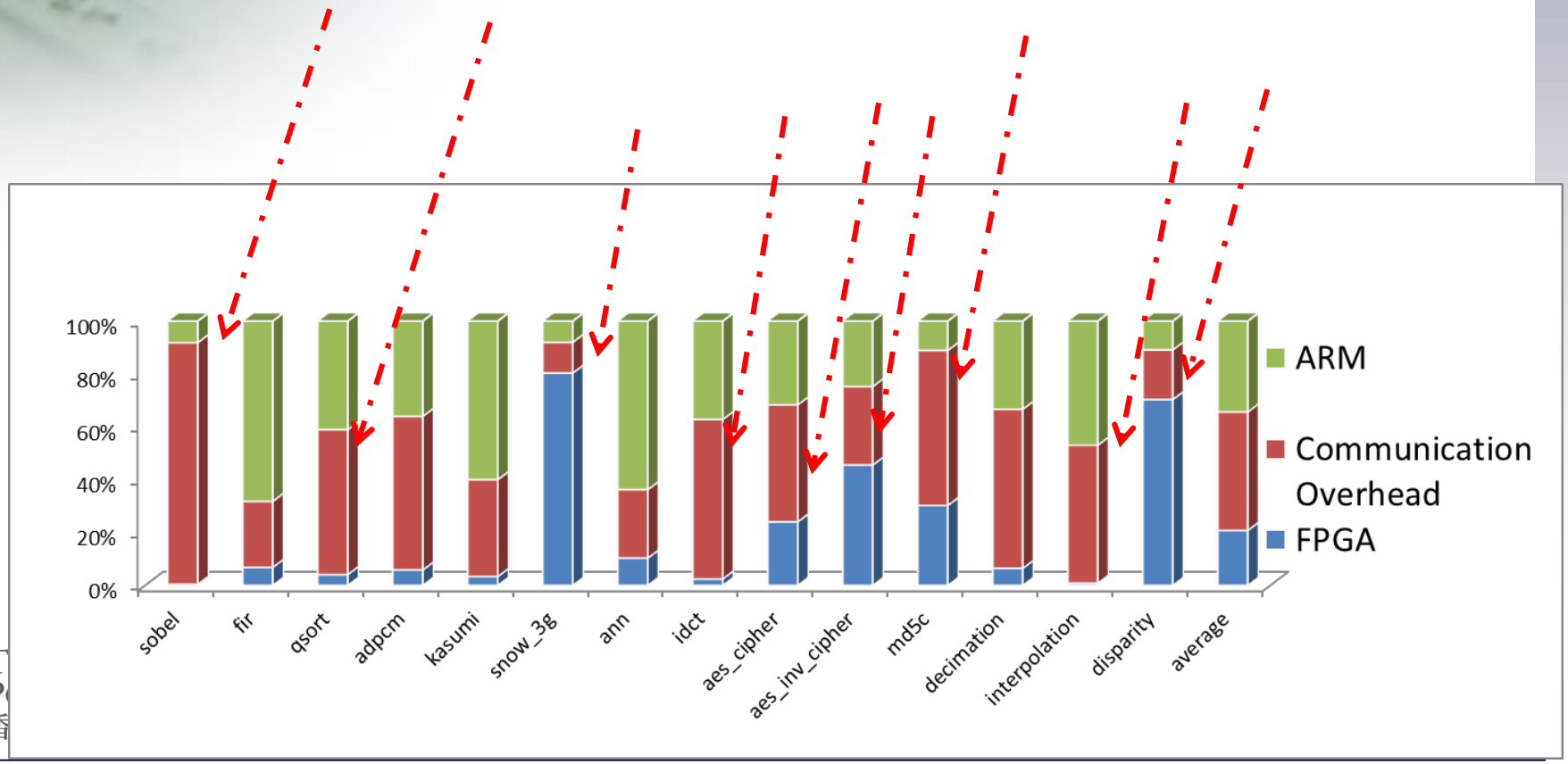




AS2CBench Result

Spend long time sending and receiving data and hence the communication overhead weights down any potential speed-up.

There are four notable exceptions:
snow3G, aes (cipher and decipher) and the disparity estimator.

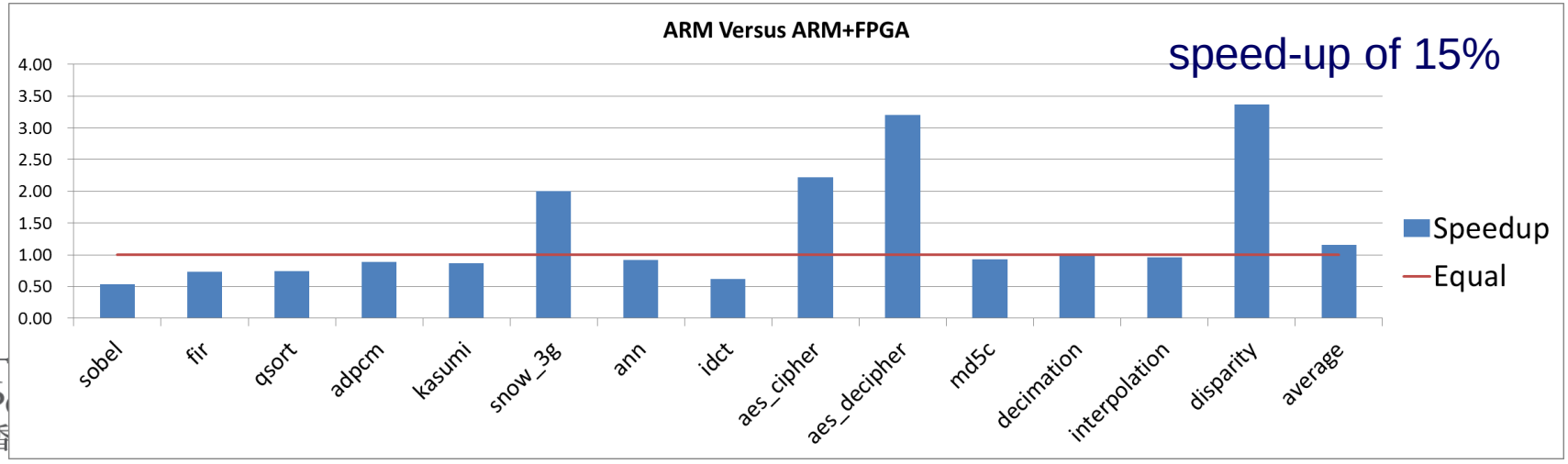


AS2CBench Result

looks closer at the comparison between these two versions (ARM vs. ARM+FPGA)

- The salient point here is that for smaller designs with very small latencies, the pure SW version achieves better results and hence it does not make sense to have an accelerated HW version.

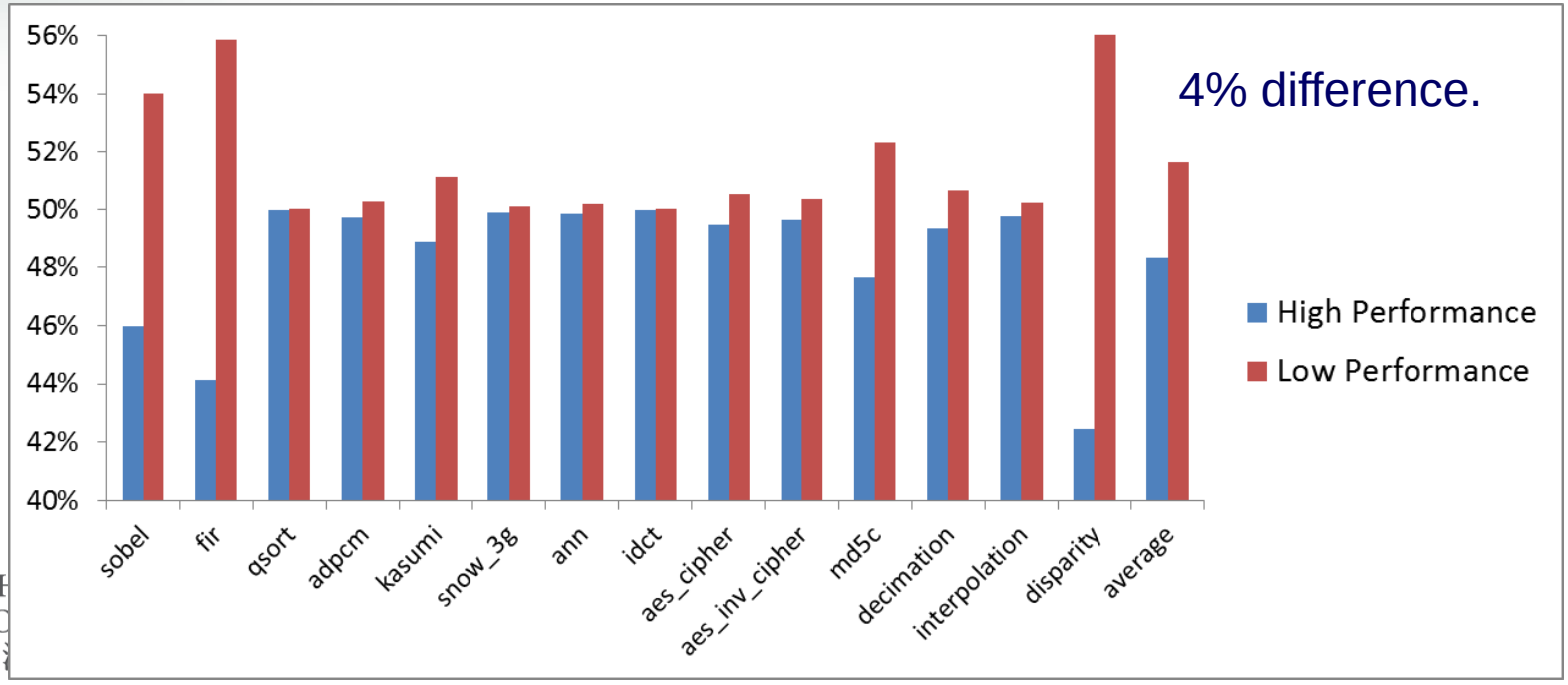
E.g. the aes has to perform 9 times the subbyte, shiftrows, mixcolumns and addroundkey functions before returning the result to the processor.



AS2CBench Result

HLS >>> different area vs. performance trade-offs

without having to modify the behavioral description.

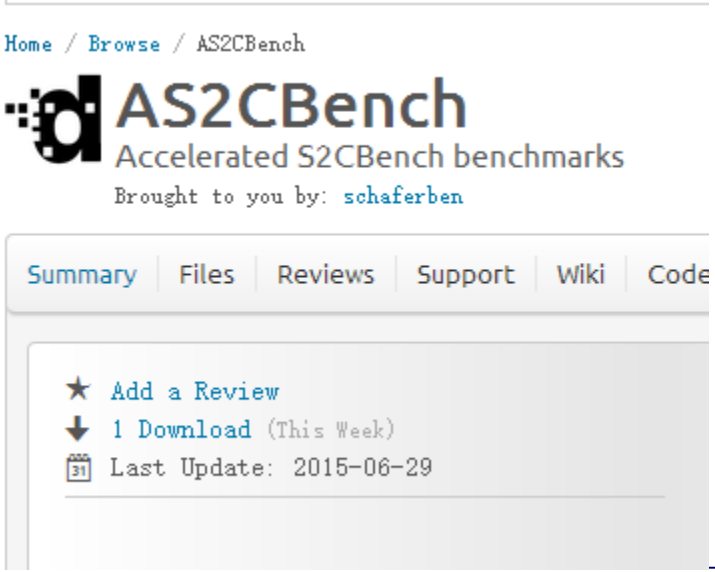


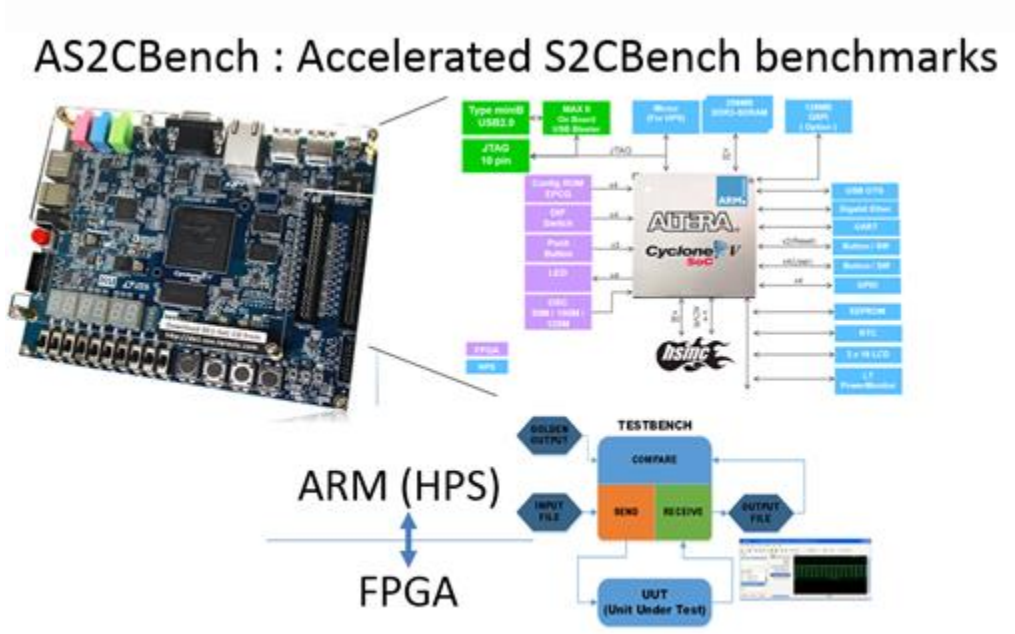
AS2CBench

The source code is open source and fully available at [2],

- ◆FPGA configuration file (SOF)
- ◆pre-compiled SW program (EXE),

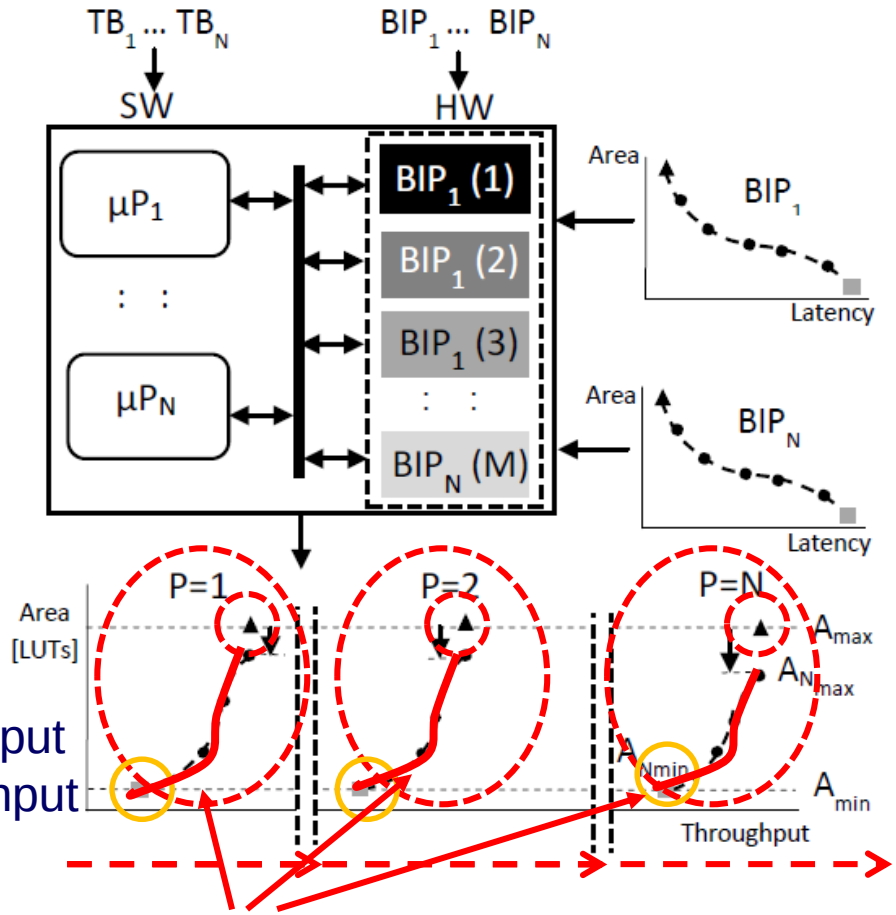
which allow to immediately have a HW/SW co-designing system up and running.





In Situ CSoCs-Based DSE

◆Motivation and Target Platform



Observation 1:
Using less processors should lead to systems with lower overall throughput.

Observation 2:
lowest latency (All case) – High Throughput
Largest latency (All case) – Low Throughput

The main aim of the work is to find a trade-off curve of optimal multiprocessor systems

In Situ CSoCs-Based DSE

Different Mapping

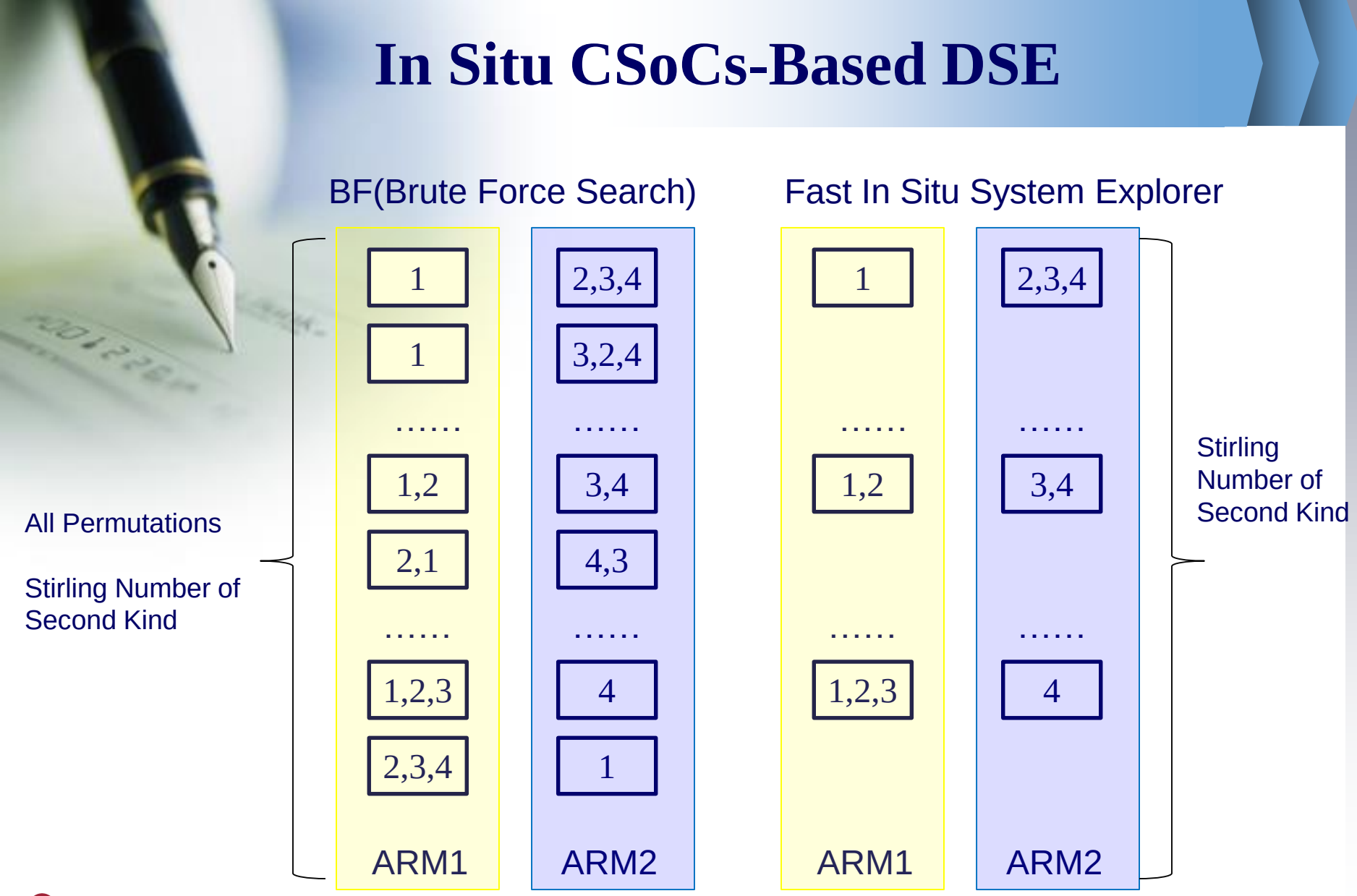
Numbers of mappings in each case can be calculated as [35]:

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^n$$

	Masters (processors) P			
Mappings	1	2	3	4
Combinations	{(1, 2, 3, 4)}	{(1), (2, 3, 4)} {(2), (1, 3, 4)} {(3), (1, 2, 4)} {(4), (1, 2, 3)} {(1, 2), (3, 4)} {(1, 3), (2, 4)} {(1, 4), (2, 3)}	{(1, 2), (3), (4)} {(1, 3), (2), (4)} {(1, 4), (2), (3)} {(1), (2, 3), (4)} {(1), (2, 4), (3)} {(1), (2), (3, 4)}	{(1), (2), (3), (4)}

Mapping : 1 7 6 1

In Situ CSoCs-Based DSE



In Situ CSoCs-Based DSE

Every optimal solution ⇒ 1. All possible permutations
2. All different Mapping

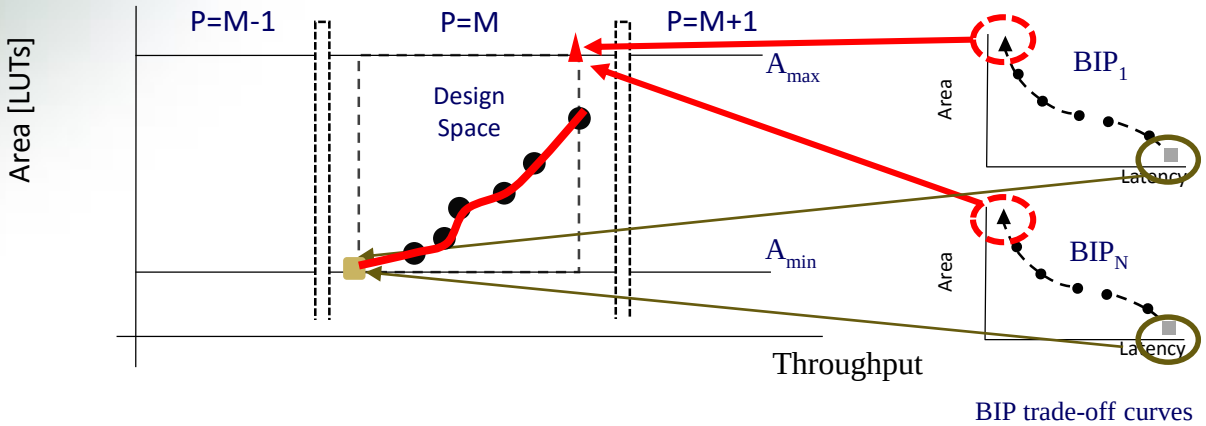
↓

periodically repeating task execution

↓

The sterling number of second kind mappings is enough

Exploration



BIP trade-off curves

This step continues until the smallest configuration with the same performance of the configuration composed of the largest, but fastest designs of each BIP is found.

Result

◆HLS tool : CyberWorkBench v.5.4 from NEC

When synthesized in Quartus II 15.0,
The Logic utilization (in ALMs) for the largest system is :
30,989 / 32,070 (97 %)

◆Altera
Cyclone V SoC

--- dual-core ARM
Cortex A-9 processor
(Ubuntu 32Bit Machine)

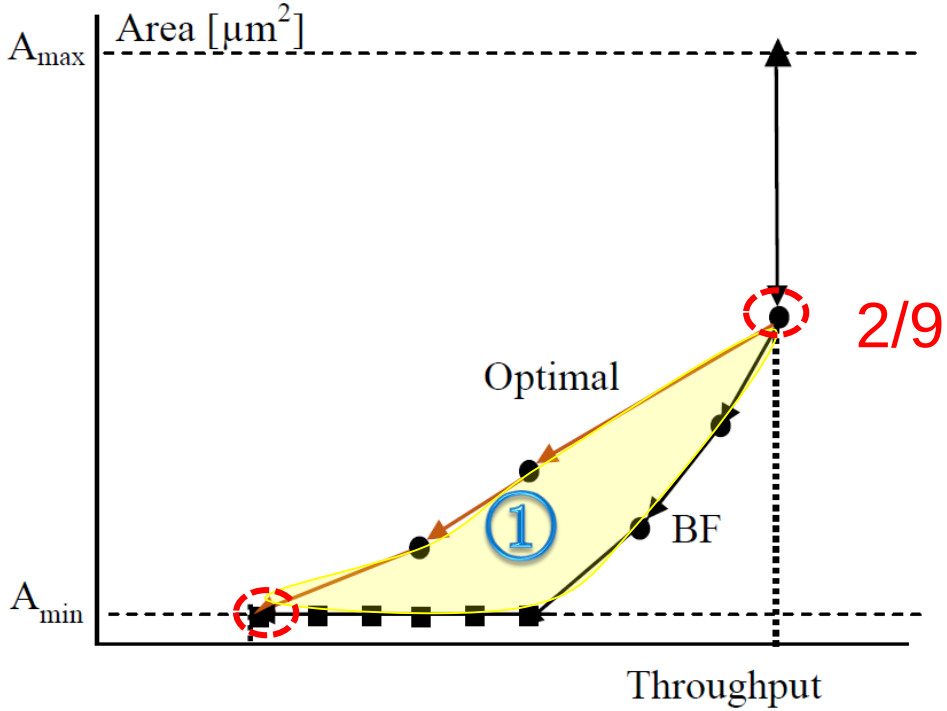
Table 6.2: Complex System Benchmark

Bench	DSE	S1	S2	S3	S4	S5	S6	S7	S8
aes_cipher	3	1	1		1	1		1	1
qsort	3			1		1	1	1	1
interp	3		1		1		1	1	1
aes_decipher	3	1		1	1	1	1		1
Task		2	2	2	3	3	3	3	4
Design		6	6	6	9	9	9	9	12

◆-- Aes_cipher is the Advanced Encryption Standard algorithm
◆-- Aes_decipher the aes decryption part of aes.
◆-- Qsort is a quick sort algorithm
◆-- interp a 4 stage interpolation filter.

Result

1. *Hypervolume:* The smaller the value, the higher the quality of the result is.
2. *Pareto Dominance: This index is equal to the ratio between the total* number of points in the Pareto set being evaluated, also present in th reference Pareto set. The higher the value, the better the Pareto set is.



Result

-- up to 42% for the *BF case*
-- *up* to 33% for our fast method,
while on average by 23% and 20% respectively.

Table 6.3: Quality of Reusults (QoR) Comparison between Exhaustive Search and Fast In Situ System Explorer (FISSE)

	S1		S2		S3		S4		
Masters	1	2	1	2	1	2	1	2	
BF	0.77	0.77	0.92	0.94	0.67	0.67	0.81	0.79	
FISSE	0.77	0.97	0.92	0.97	0.67	0.67	0.81	0.81	
Dominance	4/4	4/5	3/4	4/5	3/3	3/3	6/6	3/4	
Hypervolume	0.03	0.01	-0.06	0.10	-0.12	0.00	0.00	0.03	
	S5		S6		S7		S8		**Avg**
Masters	1	2	1	2	1	2	1	2	
BF	0.72	0.58	0.70	0.70	0.78	0.95	0.73	0.85	**0.77**
FISSE	0.72	0.72	0.70	0.70	0.78	0.99	0.73	0.87	**0.80**
Dominance	10/10	5/6	7/7	7/8	9/12	7/9	7/9	7/11	**86/100**
Hypervolume	-0.01	0.00	-0.02	0.05	-0.02	0.06	0.00	0.02	**0.004**

Result

on average only 3% worse while on average 15× faster,

showing that it can lead to very good results quickly.

Table 6.4: Running Time Comparison [min]

Bench	S1	S2	S3	S4	S5	S6	S7	S8	Avg
BF	1.32	0.91	1.00	21.71	14.75	12.75	11.32	346.6	**51.30**
FISSE	0.25	0.30	0.24	2.17	1.13	1.35	1.59	21.8	**3.60**
SpUp	5.4	3.0	4.3	10.0	13.0	9.5	7.1	15.9	**14.3**

Result

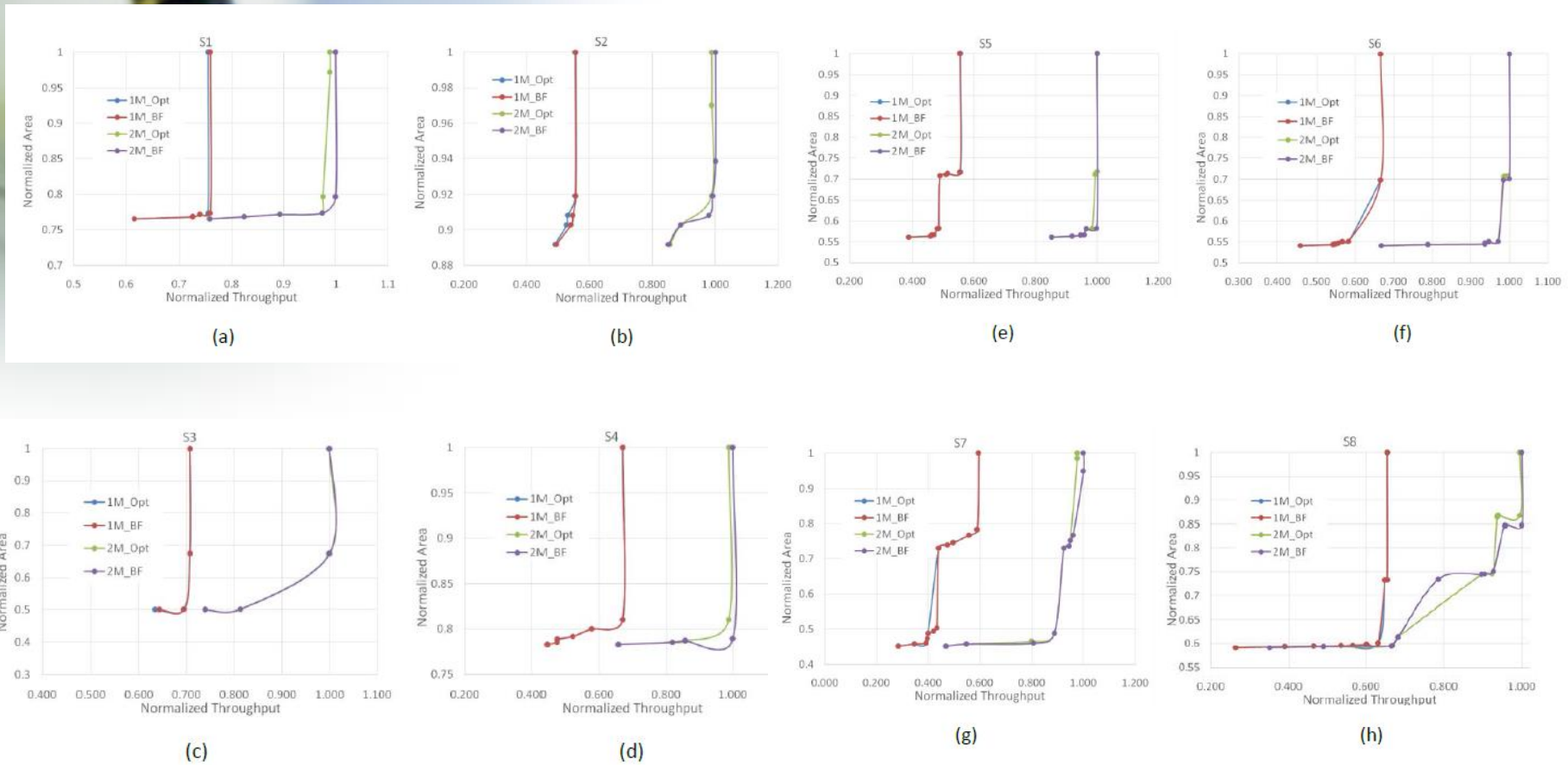


Fig. 6.4: System Exploration trade-off curves for each benchmark for 1 processor (*P=1) and 2 processors (P=2) comparing the brute force (BF) and our* proposed fast heuristic (*FISSE).* *Page 49*

Conclusions and Future Work

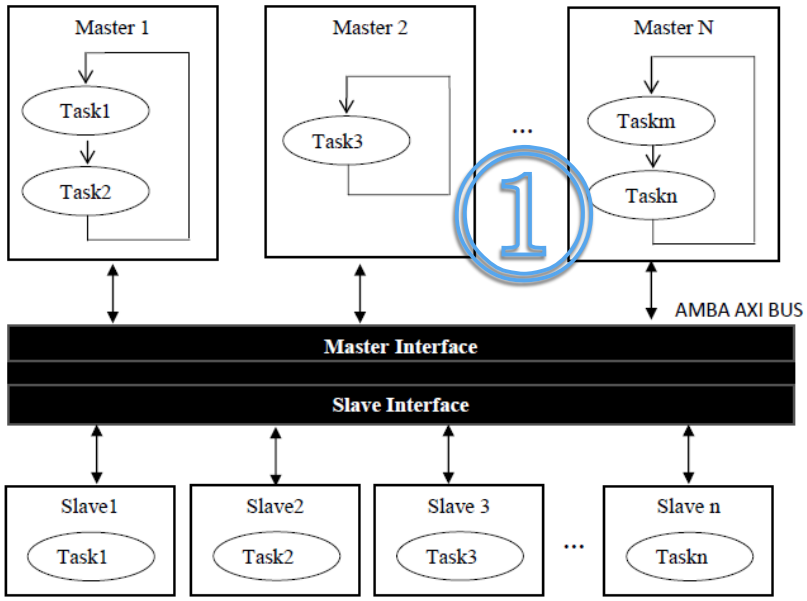
- This thesis first presented an accelerated version of the S2CBench benchmark suite to experiment on HW/SW co-design for configurable SoCs(CSoC).
- Secondly, and the main contribution, is the development of a fast method to characterize complete HW/SW systems mapped onto these CSOCs using Behaviors IPs as slaves to accelerated different tasks.

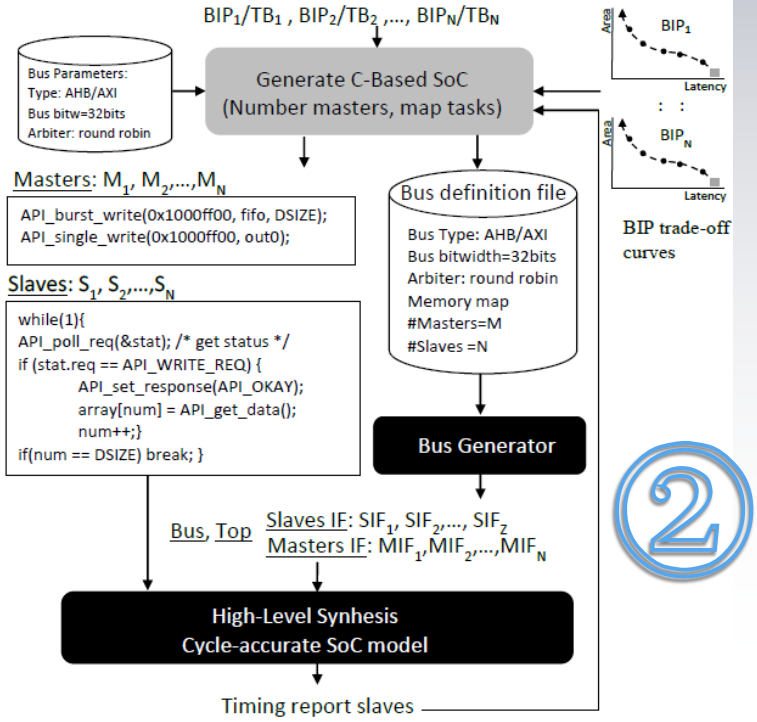
Conclusions and Future Work

◆Future work

① larger CSoC with more than 2 HPS

② Comparing the efficiency of our proposed method compared to a offline simulation.





Publication

- S. XU, B. Carrion Schafer, "AS2CBench:Accelerated Synthesizable SystemC Benchmark Suite for HW/SW Co-design", under review.
- S. XU, Y. Liu, B. Carrion Schafer, "In-situ C-based Configuration SoCs Design Space Exploration", submitted.

Reference

◆Refer to the Thesis P.55-P.56

[1] B. Carrion Schafer and A. Mahapatra, „S2cbench: Synthesizable systemc benchmark suite for high-level synthesis“, *Embedded Systems Letters, IEEE*, vol. 6, no. 3, pp. 53–56, 2014.

[2] The Hong Kong Polytechnic University DARClab. (2015), [Online]. Available: `http://sourceforge.net/projects/as2cbench/`.

[3] Moore's law. (). Moore, [Online]. Available: `https://en.wikipedia.org/wiki/Moore%27s_law`.

[4] I. T. R. for Semiconductors, „Www.public.itrs.net/links/2013itrs/2013chapters“, 2013.

[5] Accellera. (2015). Http://accellera.org, [Online]. Available: {`http://accellera.org/images/downloads/drafts-review/SystemC_Synthesis_Subset_Draft_1_4.pdf`}.

.....

Thanks !