WASEDA UNIVERSITY

# EACH: An Energy-Efficient High-Level Synthesis Framework for Approximate Computing

**Cong Hao**, Takeshi Yoshimura

WASEDA University
Graduate School of Information, Production and Systems

# Outline

- **Introduction**
  - What is **High Level Synthesis?**
  - Why we need **High-Level Synthesis** for **Approximate Computing**?
- **Problem Formulation**
- **EACH: A High-Level Synthesis Framework for Approximate Computing**
  - Functional Unit Allocation
  - Operation Scheduling
- **Experimental Results & Conclusion**

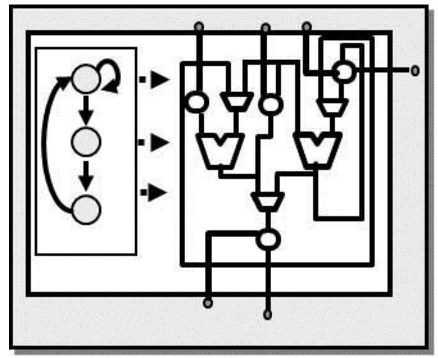
# What is High Level Synthesis (HLS)?

## VLSI Design Flow

System Specification
(Natural Language)

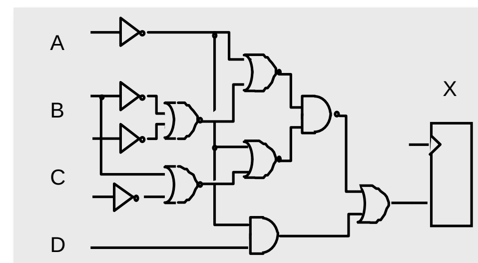
```
Process()
  wait until clk
     if and then
       x:=a+b;
       y:=a*b;
     else
       x:=a-b;
       y:=y+1;
     endif;
```

Behavioral Description
(C/C++ etc)



RTL
(verilog/VHDL)

**High-Level Synthesis**




Logic Synthesis

Gate
(verilog/VHDL)

Physical Synthesis

Mask
(GDSII)

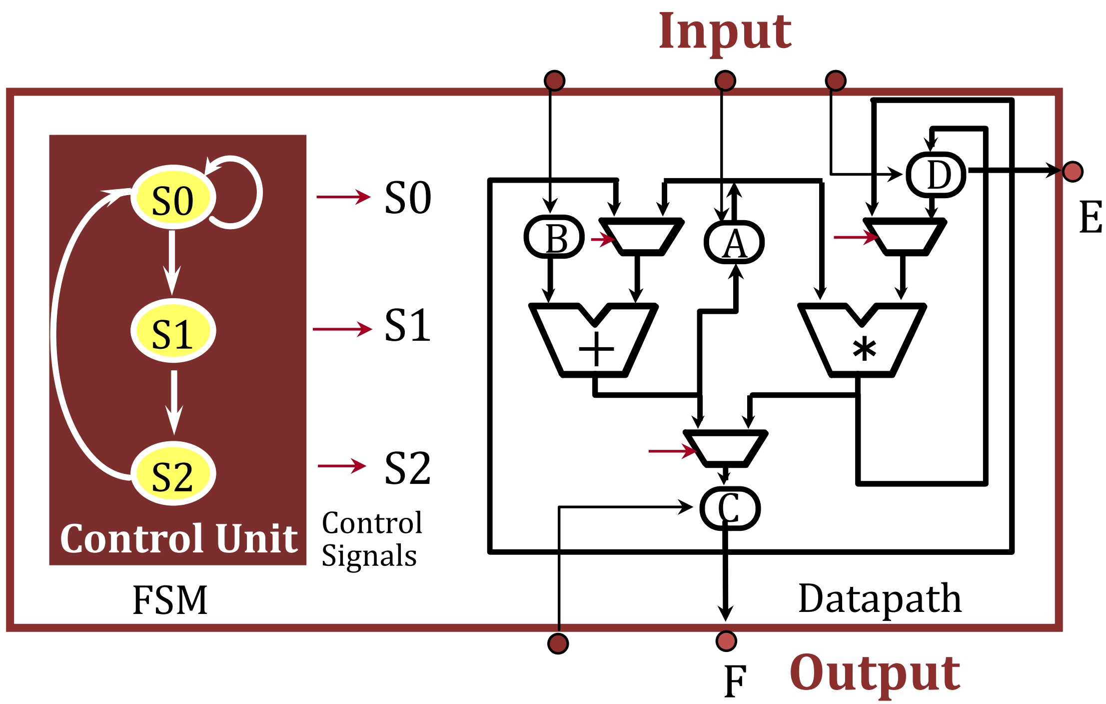

WASEDA University
Graduate School of Information, Production and Systems

# What is High Level Synthesis (HLS)?

- An automated design process from an **algorithmic description** to **a hardware that implements the algorithm**

```
Process()
 wait until clk
   if and then
     x:=a+b;
     y:=a*b;
   else
     x:=a-b;
     y:=y+1;
   endif;
```

**Algorithmic (Behavior) Description**



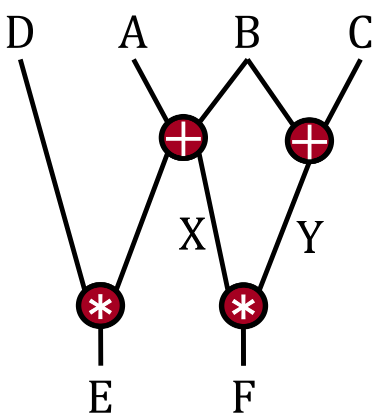

Register-Transfer-Level **Hardware Implementation**

# Scheduling and Binding

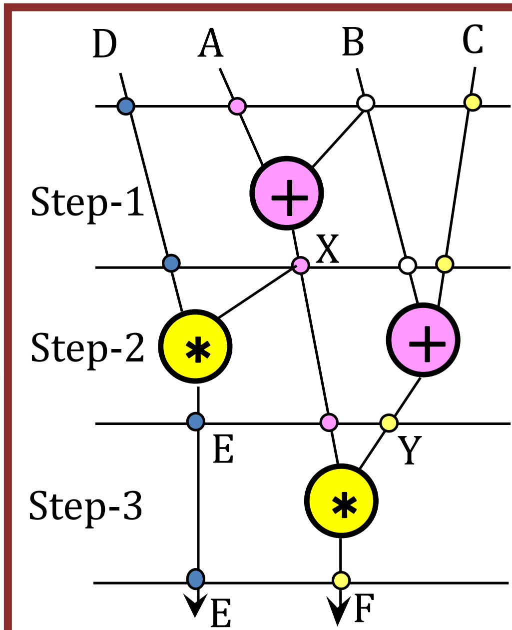
- Two major steps in High-Level Synthesis

**X = A + B;**
**E = X * D;**
**F = (B + C) * X;**

Behavior Description

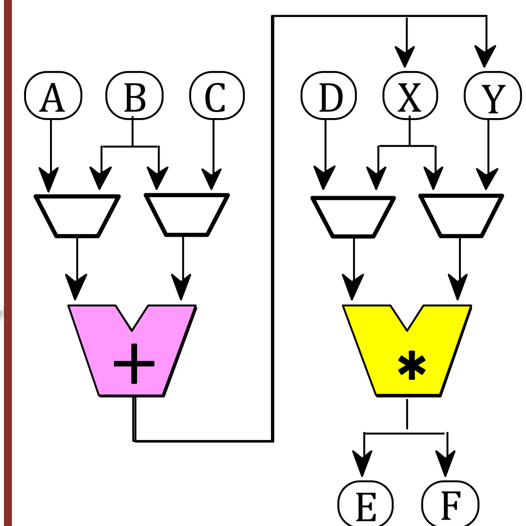



**Data Flow Graph (DFG)**




Adder x1
Multiplier x1

**Scheduling**




Register x 8
Multiplexer x 4

**Binding**

# Outline

- **Introduction**
  - What is **High Level Synthesis?**
  - Why we need **High-Level Synthesis** for **Approximate Computing**?
- **Problem Formulation**
- **EACH: A High-Level Synthesis Framework for Approximate Computing**
  - Functional Unit Allocation
  - Operation Scheduling
- **Experimental Results & Conclusion**

# HLS for Approximate Computing?

- **Too complex for manual approximate circuit design**
  - Large circuit scale
  - Manual control of the design parameters and specifications is difficult
- **There is not still a well-established methodology for automated construction of approximate systems and circuits**
- **Mechanisms to pass application intent to physical implementation flow need to be developed ***

*Swann, Gavin, Martha Prevezer, and David Stout. *The dynamics of industrial clustering: International comparisons in computing and biotechnology*. Oxford University Press, 1998.

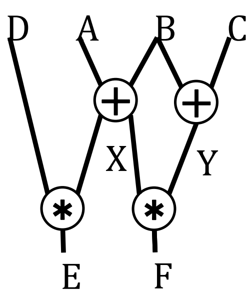
# Outline

- Introduction
- **Problem Formulation**
  - **Inputs** and **Outputs**
  - **Error Control** – the **constraints**
  - **Energy** Consumption – the **objective**
- EACH: A High-Level Synthesis Framework for Approximate Computing
- Experimental Results & Conclusion

# Problem Formulation – Inputs and Outputs

- **Inputs**
  - Data Flow Graph (**DFG**)
  - Different implementations of **functional units** (**FU**) with various design specifications (**power**, **accuracy**, **area**, etc.)
- **Outputs**

  A **scheduled DFG** where each operation is specified with an **accurate or approximate FU** implementation

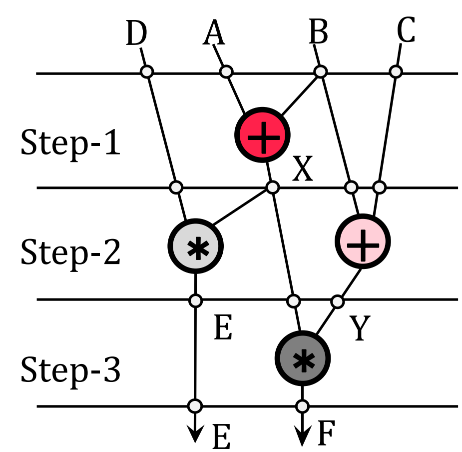



**DFG**

**Power** **Error**

| | | Area ($\mu m^2$) | Delay ($nm$) | Power($mW$) Dynamic | Leakage | Error Prob. | Mean | Max | Var. |
|---|---|---|---|---|---|---|---|---|---|
| ADD | Precise | 41 | 10 | 28 | 18 | - | - | - | - |
| | Appr. 1 | 22 | 8 | 20 | 14 | 0.15 | 3.5% | 57.14% | 42 |
| | Appr. 2 | 24 | 8 | 16 | 14 | 0.2 | 4.5% | 57.14% | 6833 |
| | Appr. 3 | 14 | 6 | 10 | 10 | 0.23 | 6.0% | 57.14% | 10905 |
| MUL | Precise | 710 | 22 | 122 | 83 | - | - | - | - |
| | Appr. | 468 | 14 | 68 | 82 | 0.81 | 3.32% | 22.22% | $4.3\times10^5$ |

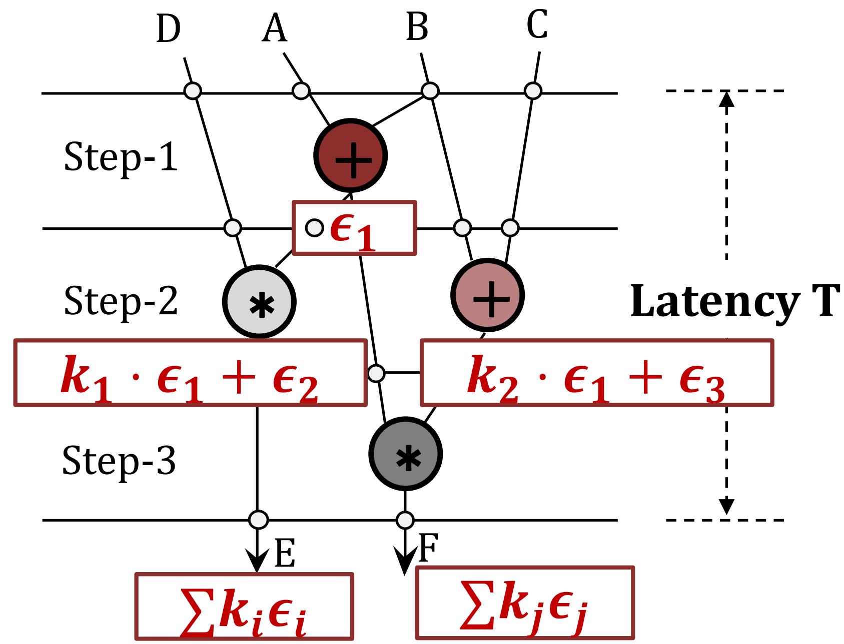
**FU implementations**




**A scheduled DFG with determined FU implementations**

# Problem Formulation – Constraints

- **Latency Constraint**
  - General constraint for HLS
- **Output Error Constraints**
  - **Special for Approximate Computing**
  - **Error Propagation**

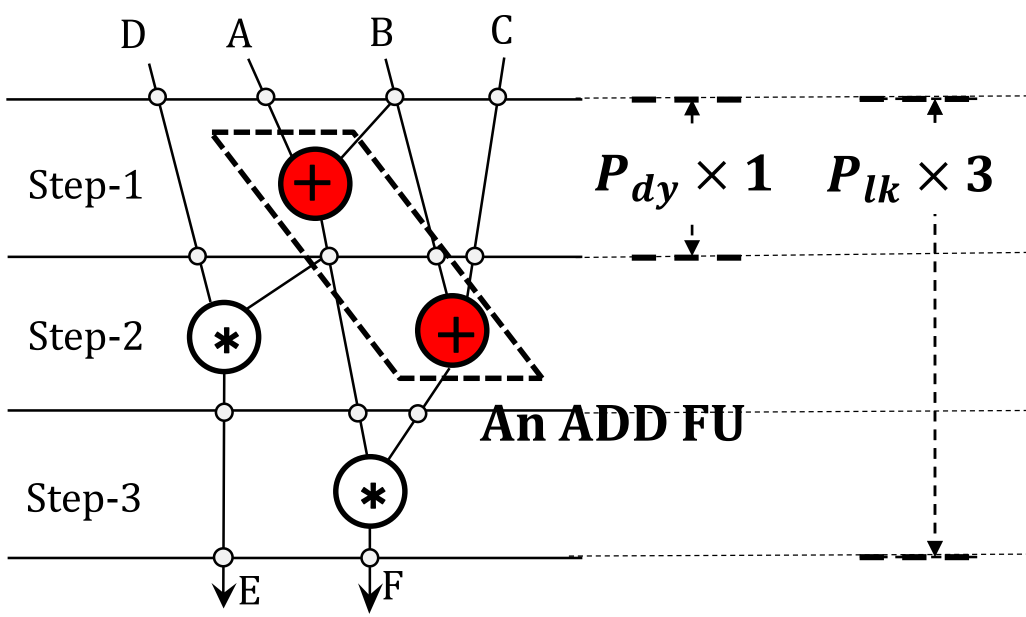



**Output errors need to be bounded**

| | | $o_1$ | $o_2$ | $\cdots$ | $o_n$ | |
|---|---|---|---|---|---|---|
| $op_1$ type $\tau_1$ | $F_{\tau_1}^{\varphi_1}$ | $vs_{op_1,o_1}^{1}$ | $vs_{op_1,o_2}^{1}$ | $\cdots$ | $vs_{op_1,o_n}^{1}$ | $EV_{op_1}^{1}$ |
| | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | |
| | $F_{\tau_1}^{\varphi_k}$ | $vs_{op_1,o_1}^{k}$ | $vs_{op_1,o_2}^{k}$ | $\cdots$ | $vs_{op_1,o_n}^{k}$ | $EV_{op_1}^{k}$ |
| $op_2$ type $\tau_2$ | $F_{\tau_2}^{\varphi_1}$ | $vs_{op_2,o_1}^{1}$ | $vs_{op_2,o_2}^{1}$ | $\cdots$ | $vs_{op_2,o_n}^{1}$ | $EV_{op_2}^{1}$ |
| | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | |
| | $F_{\tau_2}^{\varphi_k}$ | $vs_{op_2,o_1}^{k}$ | $vs_{op_2,o_2}^{k}$ | $\cdots$ | $vs_{op_2,o_n}^{k}$ | $EV_{op_2}^{k}$ |
| | | $\cdots$ | | | | |
| *system error constraint* | | $v_B(o_1)$ | $v_B(o_2)$ | $\cdots$ | $v_B(o_n)$ | |

# Problem Formulation – Objective

- **Minimize the energy consumption**
  - Dynamic power $P_{dy}$
  - **Leakage power $P_{lk}$** – dominated by **resource usage**




Energy consumption for the ADD FU:

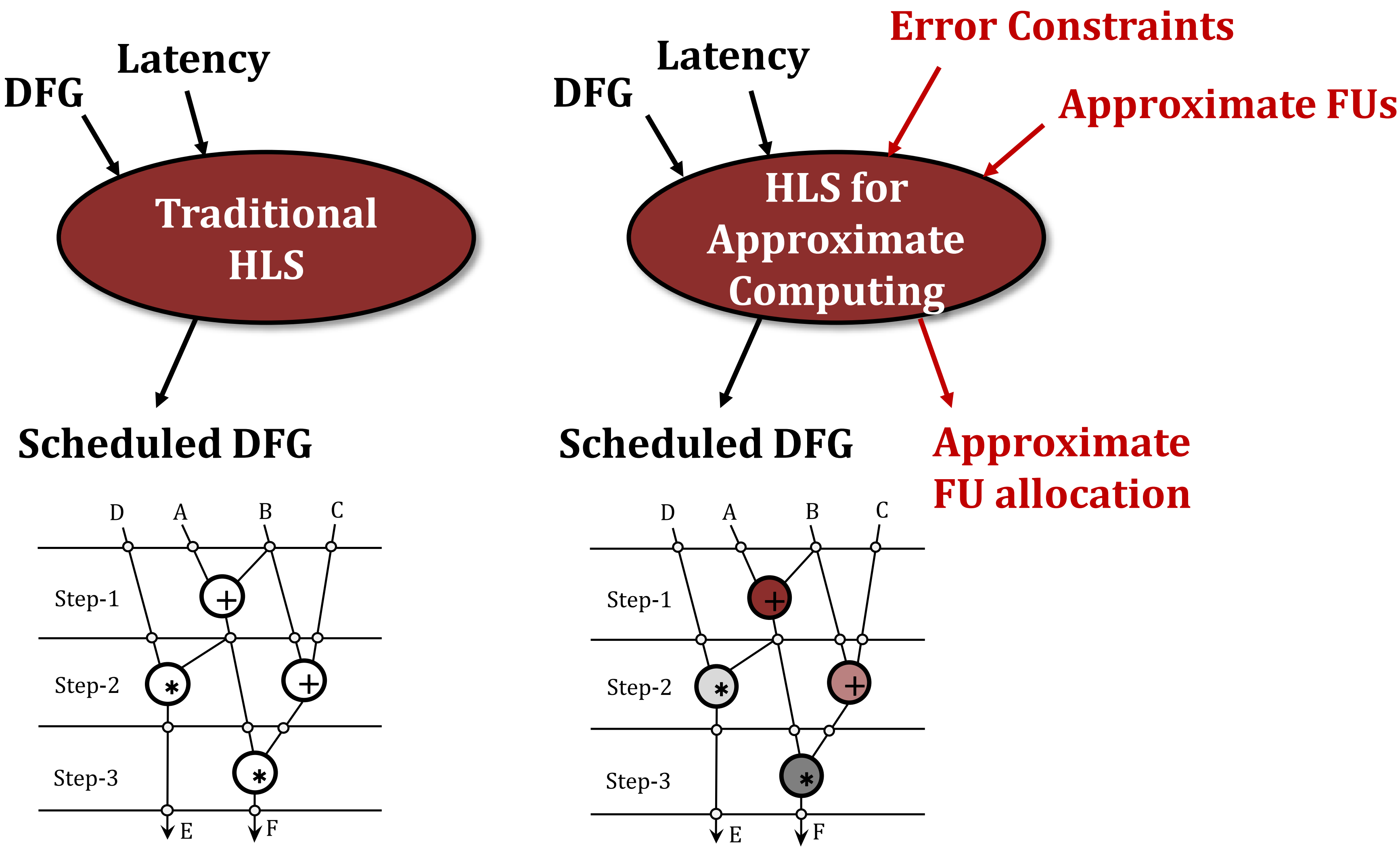
$$P_{dy} \times 1 + P_{lk} \times 3$$

Dynamic Energy (↑ $P_{dy} \times 1$)

Leakage Energy (↑ $P_{lk} \times 3$)

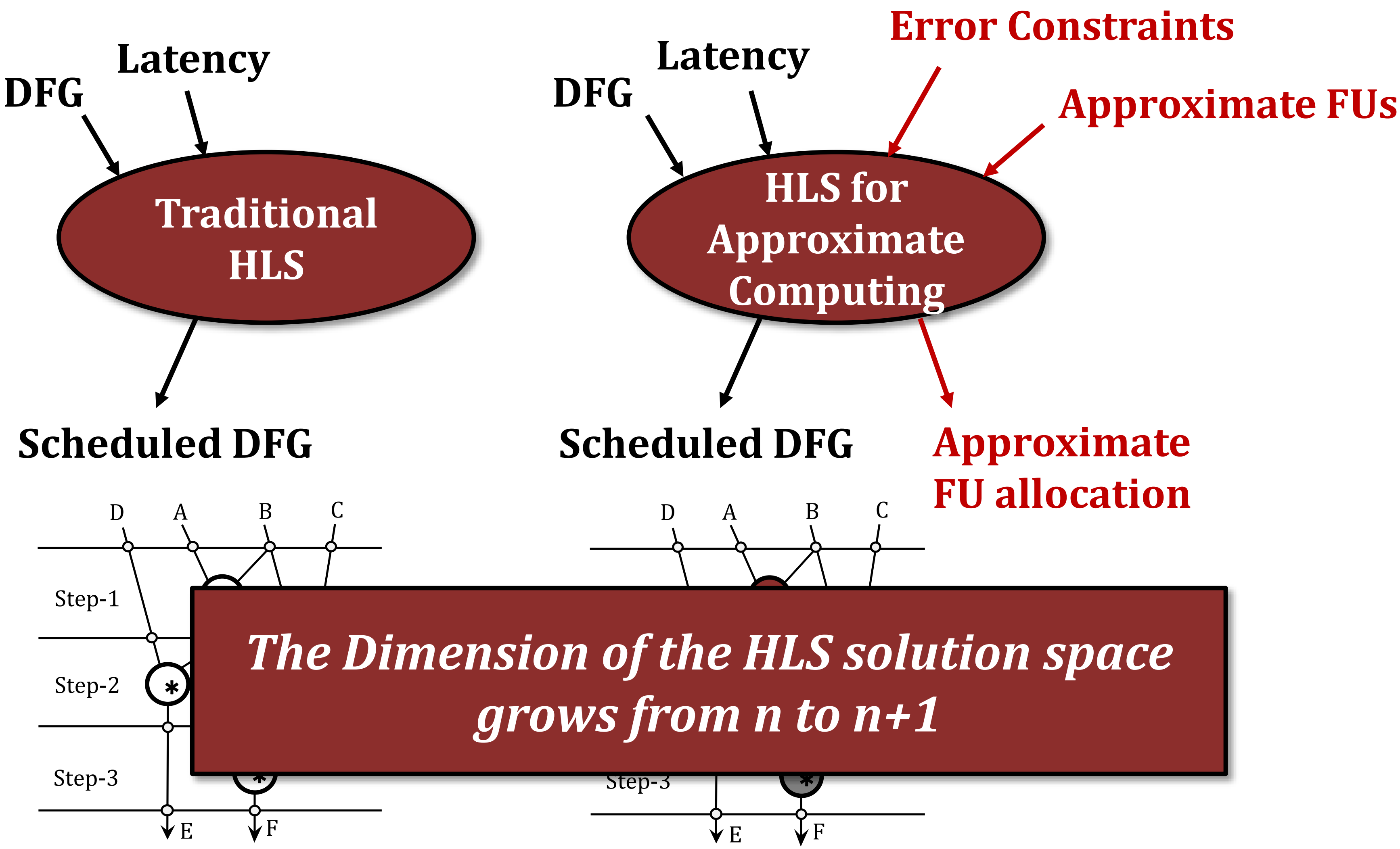
# Brief Summary

- **Traditional HLS vs. HLS for Approximate Computing**



# Brief Summary

- <u>Traditional HLS</u> vs. <u>HLS for Approximate Computing</u>



# Outline

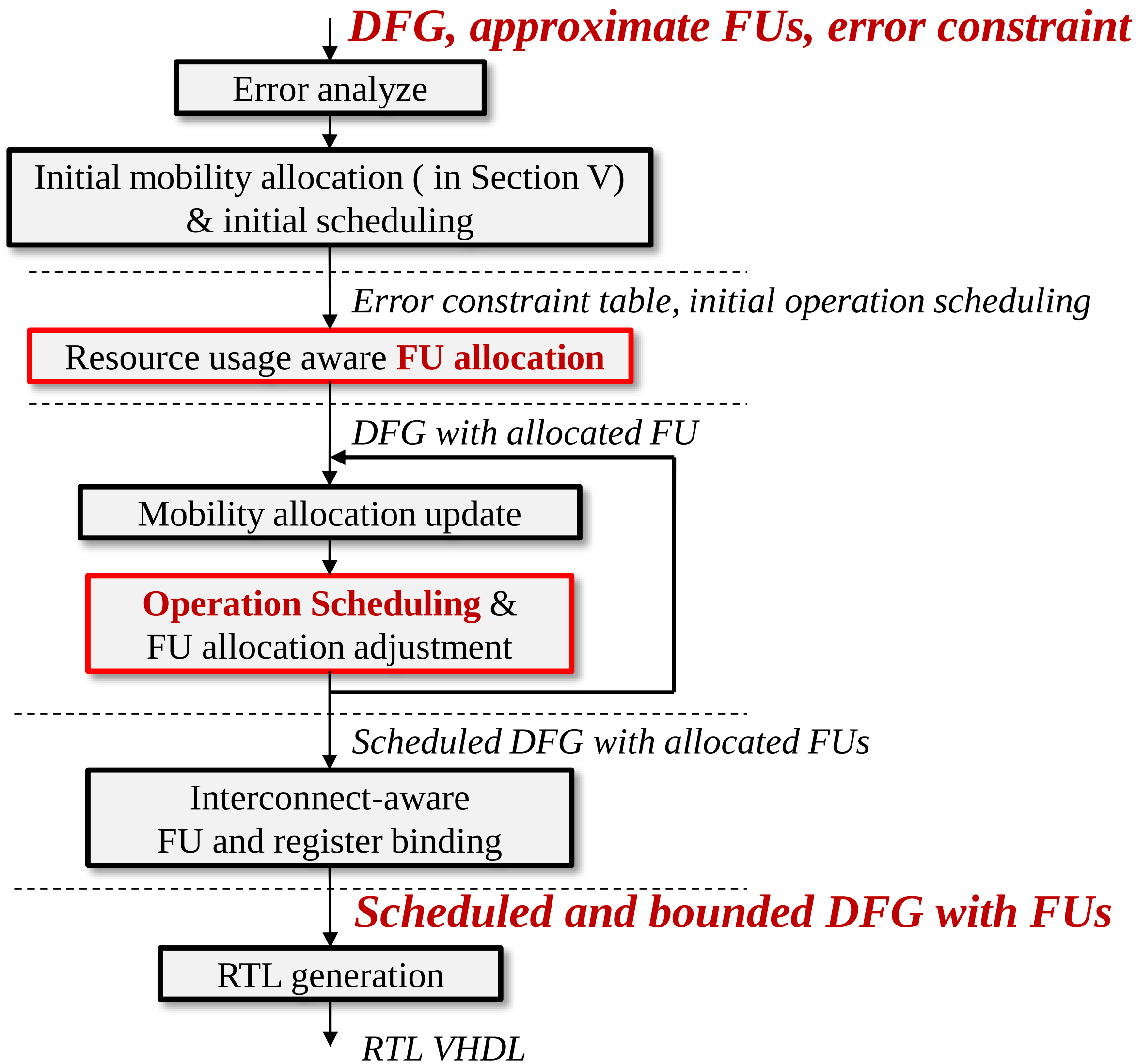
- Introduction
- Problem Formulation
- **EACH: An Energy-Efficient Approximate Computing High-Level Synthesis Framework**
  - **Initial Solution**: Functional Unit Allocation
  - **Optimization**: Operation Scheduling
- Experimental Results & Conclusion

# Outline

- Introduction
- Problem Formulation
- **EACH: An Energy-Efficient Approximate Computing High-Level Synthesis Framework**
  - **Initial Solution**: Functional Unit Allocation
  - **Optimization**: Operation Scheduling
- Experimental Results & Conclusion

# Overall Flow of **EACH**

1. **Pre-processing**
2. **FU Allocation : Initial Solution**
3. **Scheduling: optimization**
4. **Binding**

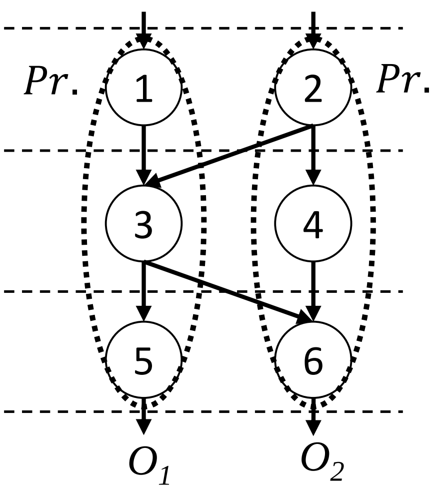


# Outline

- Introduction
- Problem Formulation
- **EACH: An Energy-Efficient Approximate Computing High-Level Synthesis Framework**
    - **Initial Solution: Functional Unit Allocation**
    - Optimization: Operation Scheduling
- Experimental Results & Conclusion

# Functional Unit (FU) Allocation

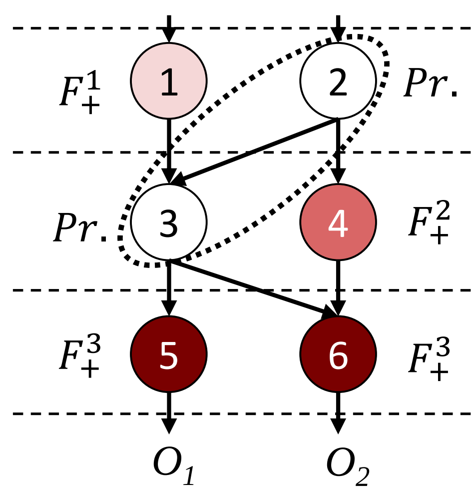
- **Objective –** Use approximate FUs as many as possible under error constraints to minimize energy
- **Previous Work and Motivating Example**
  - The Multiple-choice Multiple-dimension Knapsack (**MMKP**)
  - **Resource sharing is ignored – increased leakage energy**




**Resource : 2 FUs**
*Precise* $\times 2$

**Energy:**
$\sum Leakage + \sum Dynamic = \mathbf{276}$

**Solution with all precise FUs**



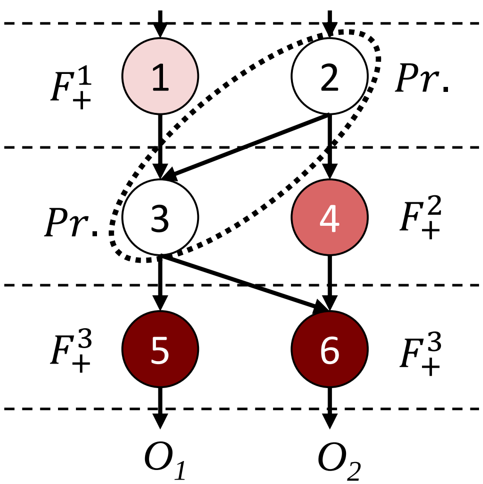

**Resource : 5 FUs**
*Precise* $\times 1$
$F_+^1 \times 1$
$F_+^2 \times 1$
$F_+^3 \times 2$

**Energy:**
$\sum Leakage + \sum Dynamic = \mathbf{310}$

**Knapsack solution**

# Functional Unit (FU) Allocation

- **Objective –** Use approximate FUs as many as possible under error constraints to minimize energy
- **Previous Work and Motivating Example**
  - The Multiple-choice Multiple-dimension Knapsack (**MMKP**)
  - **Resource sharing is ignored – increased leakage energy**
  - Not to increase the number of FUs – **binding aware**




**Resource : 5 FUs**

$Precise \times 1$

$F_+^1 \times 1$

$F_+^2 \times 1$

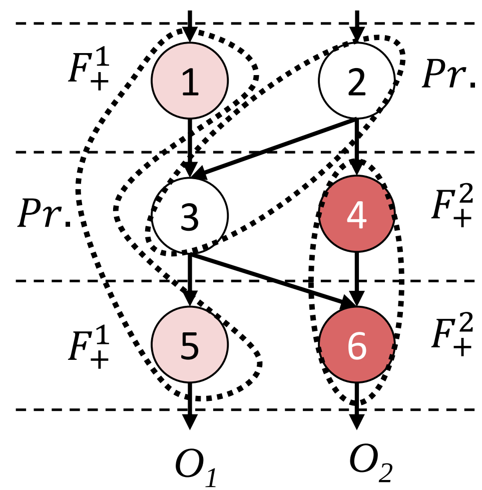
$F_+^3 \times 2$

**Energy:**

$\sum Leakage + \sum Dynamic$

**= 310**

**Knapsack solution**




**Resource : 3 FUs**

$Precise \times 1$

$F_+^1 \times 1$

$F_+^2 \times 1$

~~$F_+^3 \times 2$~~

**Energy:**

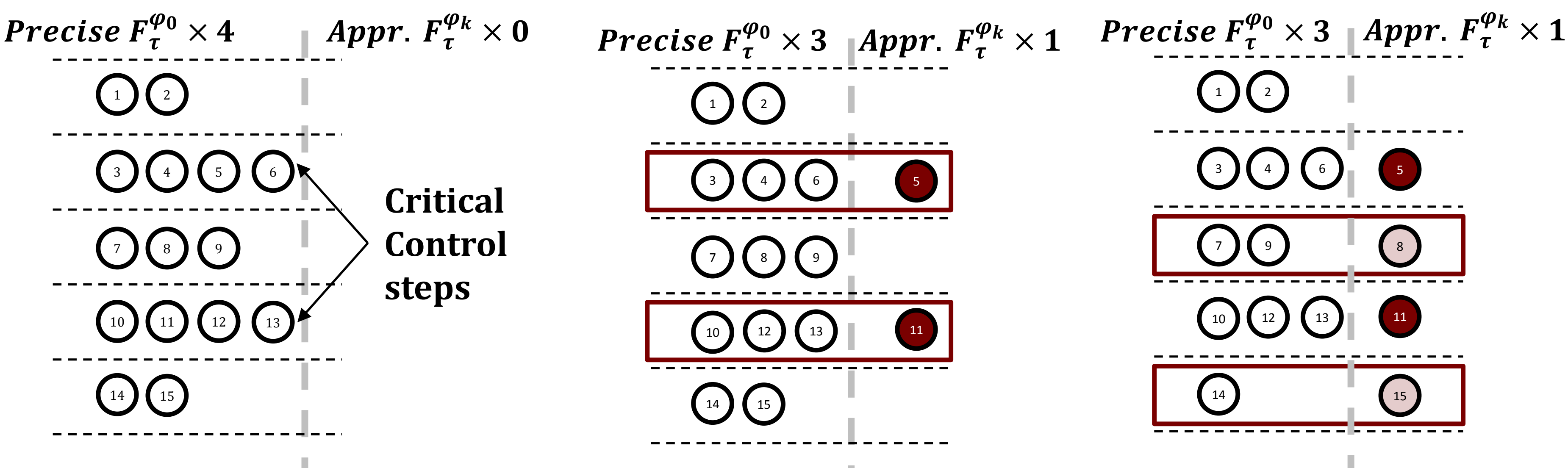
$\sum Leakage + \sum Dynamic$

**= 266**

**Binding aware solution with lower energy**

# Functional Unit (FU) Allocation

- **Proposal –** to guarantee that **the FU usage does not increase** and the **energy strictly reduces**




**1. Collect critical control steps**

The control steps that have the largest number of precise FUs

**2. Strong Replacement**

In each critical control step, replace one precise FU using a new approximate FU

**3. Weak Replacement**

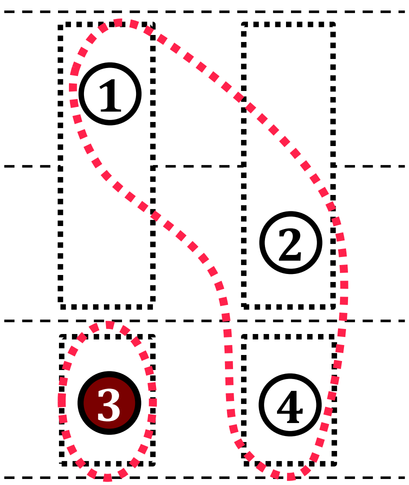
Reuse the newly allocated FU in 2 to reduce dynamic energy

# Outline

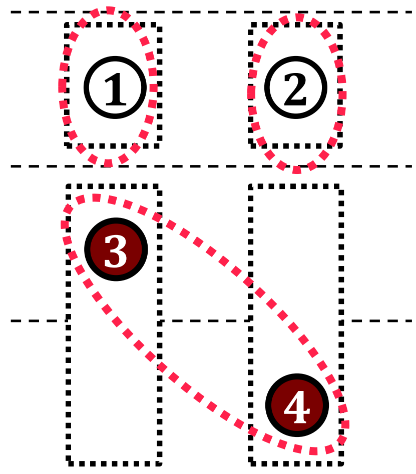
- Introduction
- Problem Formulation
- **EACH: An Energy-Efficient Approximate Computing High-Level Synthesis Framework**
  - **Initial Solution**: Functional Unit Allocation
  - **Optimization**: **Operation Scheduling**
- Experimental Results & Conclusion

# Operation Scheduling

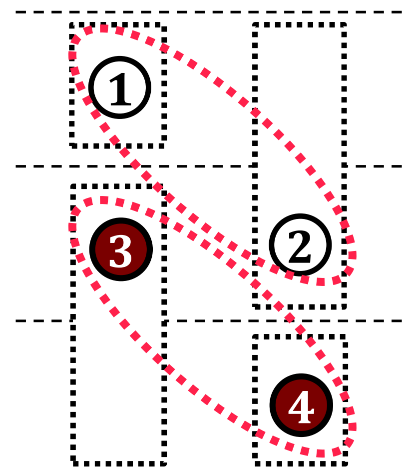
- **How to perform operation scheduling**
  - To enable as many approximate FUs are used as possible
  - To reduce total FU usage (precise, approximate)




$Appr.ADD \times 1 \quad (3)$
$Presice.ADD \times 1 \quad (1, 2, 4)$



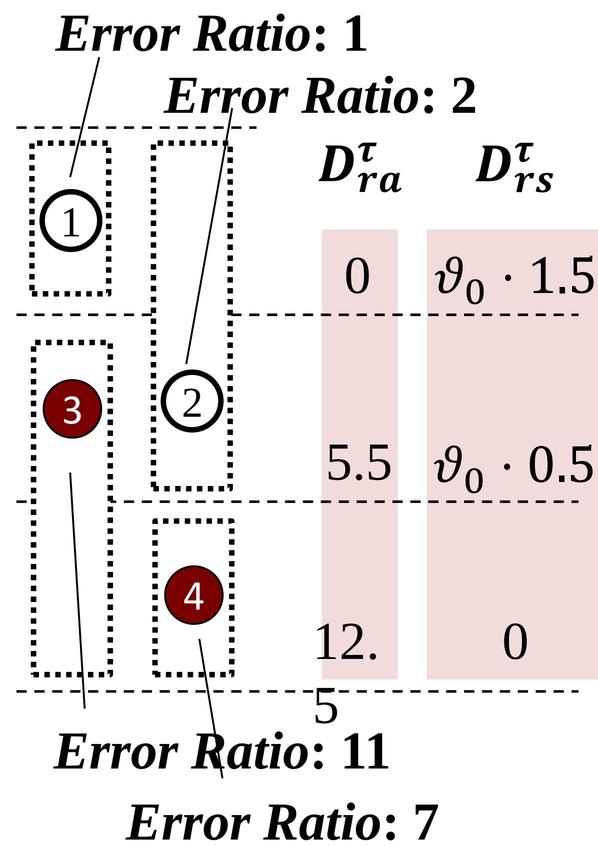

$Appr.ADD \times 1 \quad (3, 4)$
$Presice.ADD \times 2 \quad (1), (2)$




$Appr.ADD \times 1 \quad (3, 4)$
$Presice.ADD \times 1 \quad (1, 2)$

# Operation Scheduling

- **Error ratio density and resource density**
  - **Error ratio**: energy reduction per error of an operation
  - **Error radio density**: the density of operations with **large** error ratio
  - **Resource density**: the density of operations with **small** error ratio
- **Proposal –** To **uniformly distribute** the variance of error ratio density and resource density




$Appr.ADD \times 1 \quad (3, 4)$

$Presice.ADD \times 1 \quad (1, 2)$

# Outline

- Introduction
- Problem Formulation
- EACH: An Energy-Efficient Approximate Computing High-Level Synthesis Framework
  - Initial Solution: Functional Unit Allocation
  - Optimization: Operation Scheduling
- **Experimental Results & Conclusion**

# Experimental Results

- Linux RedHat with 2.8GHz CPU and 8GB memory, C language
- Error constraints are set to allow the approximate outputs be **3% to 10% different from the accurate values**
  - A commensurate error tolerance range for image processing applications

| | ar | ar2 | ad2 | arf | ellip | ewf | fft | fir | mpeg | mvd | rand0 | rand1 | rand2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of Operations | 28 | 28 | 46 | 28 | 34 | 34 | 129 | 44 | 53 | 32 | 91 | 157 | 631 |
| Error Variance* | 30 | 40 | 100 | 20 | 14 | 70 | 120 | 10 | 100 | 5 | 9 | 12 | 15 |
| Mean Error | 8.3% | 6.7% | 9.8% | 5.9% | 7.5% | 6.8% | 7.2% | 7.3% | 9.6% | 7.1% | 4.1% | 3.8% | 3.3% |

# Experimental Results

- Evaluation of **FU allocation** – **initial solution**
  - Achieved **11% total energy reduction** compared to precise circuits on average
- Evaluation of **scheduling and FU adjustment** – **optimization**
  - Achieved **another 7% energy reduction** from initial solutions

| test bench | Precise | | | KILS [14] | | | | | | | | | | | EACH(initial) | | | | | | | | | | EACH | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FU | | Energy | FU | | | | | | Energy(W) | | | | Time | FU | | | | | | Energy(W) | | | | FU | | | | | | Energy | | | | Time | |
| | $N^+$ | $N^*$ | Total(W) | $N_0^+$ | $N_1^+$ | $N_2^+$ | $N_3^+$ | $N_0^*$ | $N_1^*$ | Ely | Edy | Total | **Cmp** | (ms) | $N_0^+$ | $N_1^+$ | $N_2^+$ | $N_3^+$ | $N_0^*$ | $N_1^*$ | Ely | Edy | Total | **Cmp** | $N_0^+$ | $N_1^+$ | $N_2^+$ | $N_3^+$ | $N_0^*$ | $N_1^*$ | Ely | Edy | Total | **Cmp** | (ms) | **Cmp** |
| ar | 2 | 5 | 7.25 | 2 | 0 | 0 | 2 | 0 | 5 | 5.13 | 1.26 | 6.34 | **0.85** | 2 | 1 | 0 | 1 | 0 | 0 | 5 | 4.86 | 1.35 | 6.21 | **0.86** | 1 | 0 | 1 | 0 | 0 | 5 | 4.86 | 1.35 | 6.21 | **0.86** | 1 | 0.5 |
| ar2 | 2 | 4 | 6.34 | 0 | 1 | 0 | 2 | 0 | 4 | 3.98 | 1.22 | 5.20 | **0.82** | 2 | 0 | 0 | 1 | 1 | 0 | 4 | 3.87 | 1.24 | 5.12 | **0.81** | 0 | 0 | 1 | 1 | 0 | 4 | 3.87 | 1.24 | 5.12 | **0.81** | 2 | 1.0 |
| ad2 | 2 | 4 | 21.88 | 2 | 0 | 0 | 1 | 3 | 3 | 24.89 | 4.13 | 29.01 | **1.33** | 26 | 2 | 0 | 0 | 0 | 3 | 1 | 16.88 | 4.36 | 21.24 | **0.97** | 2 | 0 | 0 | 0 | 3 | 1 | 16.88 | 4.36 | 21.24 | **0.97** | 37 | 1.42 |
| arf | 2 | 4 | 7.44 | 2 | 0 | 0 | 2 | 0 | 4 | 5.37 | 1.35 | 6.73 | **0.90** | 2 | 2 | 0 | 0 | 0 | 0 | 4 | 5.04 | 1.39 | 6.43 | **0.86** | 1 | 1 | 0 | 0 | 0 | 4 | 4.00 | 1.42 | 5.43 | **0.73** | 2 | 1.0 |
| ellip | 2 | 4 | 12.23 | 2 | 0 | 0 | 2 | 0 | 4 | 13.20 | 2.23 | 15.43 | **1.26** | 3 | 2 | 0 | 0 | 0 | 2 | 2 | 8.78 | 2.48 | 11.26 | **0.92** | 2 | 0 | 0 | 0 | 2 | 2 | 8.78 | 2.48 | 11.26 | **0.92** | 10 | 3.3 |
| ewf | 3 | 2 | 5.66 | 1 | 1 | 1 | 3 | 0 | 2 | 3.82 | 0.84 | 4.66 | **0.82** | 8 | 1 | 1 | 0 | 1 | 0 | 2 | 3.78 | 0.96 | 4.74 | **0.84** | 1 | 0 | 1 | 1 | 0 | 2 | 3.64 | 0.93 | 4.57 | **0.81** | 4 | 0.5 |
| fft | 4 | 4 | 16.75 | 4 | 1 | 0 | 4 | 1 | 4 | 13.96 | 3.34 | 17.30 | **1.03** | 59 | 2 | 1 | 0 | 1 | 0 | 4 | 10.09 | 4.06 | 14.15 | **0.84** | 2 | 0 | 0 | 2 | 0 | 4 | 9.98 | 3.80 | 13.78 | **0.82** | 23 | 0.4 |
| fir | 3 | 3 | 10.67 | 3 | 0 | 0 | 0 | 2 | 2 | 7.17 | 2.75 | 9.92 | **0.93** | 3 | 2 | 1 | 0 | 0 | 2 | 1 | 7.15 | 2.88 | 10.04 | **0.94** | 1 | 0 | 0 | 2 | 1 | 2 | 5.36 | 2.93 | 6.49 | **0.61** | 4 | 1.3 |
| mpeg | 3 | 3 | 13.14 | 3 | 0 | 0 | 2 | 3 | 0 | 11.63 | 2.04 | 13.67 | **1.04** | 295 | 3 | 0 | 0 | 0 | 2 | 1 | 10.87 | 2.18 | 13.05 | **0.99** | 3 | 0 | 0 | 0 | 2 | 1 | 10.87 | 2.18 | 13.05 | **0.99** | 288 | 0.9 |
| mvd | 3 | 3 | 6.15 | 2 | 1 | 0 | 1 | 1 | 3 | 4.29 | 1.54 | 5.83 | **0.95** | 3 | 2 | 1 | 0 | 0 | 1 | 2 | 3.86 | 1.64 | 5.50 | **0.89** | 1 | 1 | 1 | 0 | 1 | 2 | 3.40 | 1.67 | 5.07 | **0.82** | 2 | 0.7 |
| rand0 | 7 | 6 | 14.13 | 5 | 1 | 0 | 7 | 1 | 6 | 11.98 | 2.53 | 14.51 | **1.03** | 102 | 4 | 0 | 2 | 1 | 1 | 5 | 9.65 | 2.88 | 12.53 | **0.89** | 2 | 0 | 1 | 3 | 1 | 5 | 8.21 | 2.67 | 10.88 | **0.77** | 59 | 0.6 |
| rand1 | 12 | 6 | 19.13 | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | 9 | 1 | 1 | 1 | 2 | 4 | 12.47 | 4.97 | 17.44 | **0.91** | 7 | 1 | 0 | 4 | 1 | 5 | 10.64 | 4.42 | 15.05 | **0.79** | 38 | NA |
| rand2 | 32 | 17 | 84.88 | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | 19 | 1 | 1 | 11 | 4 | 13 | 53.24 | 17.46 | 70.71 | **0.83** | 14 | 5 | 6 | 7 | 2 | 15 | 41.62 | 20.66 | 62.28 | **0.73** | 185 | NA |
| **AVR** | | | | | | | | | | | | | **0.98** | | | | | | | | | | | **0.89** | | | | | | | | | | **0.82** | | **0.89** |

**KILS: Li, Chaofan, et al. Joint precision optimization and high level synthesis for approximate computing. DAC, 2015*

# Conclusion

- A Framework of **High-Level Synthesis** for **Approximate Computing**

- **Two sub-problems:**
  - FU allocation – initial solution
  - Operation Scheduling and FU allocation adjustment – optimization

- Total **18% energy reduction** is achieved, while previous work KILS achieves 2% in average