Timing Driven Power Gating in High-Level Synthesis

Shih-Hsu Huang and Chun-Hua Cheng

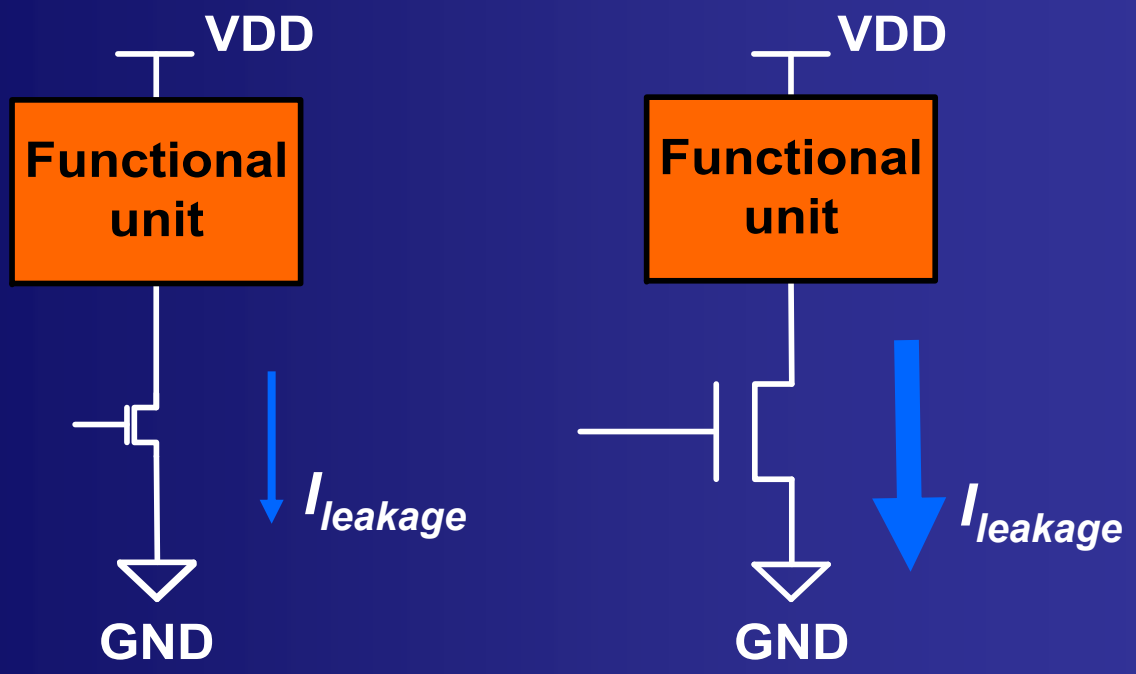
Department of Electronic Engineering
Chung Yuan Christian University, Taiwan

Outline

- Introduction
- Motivation
- Our Approach
  - MILP
  - Heuristic Algorithm
- Experimental Result
- Conclusion

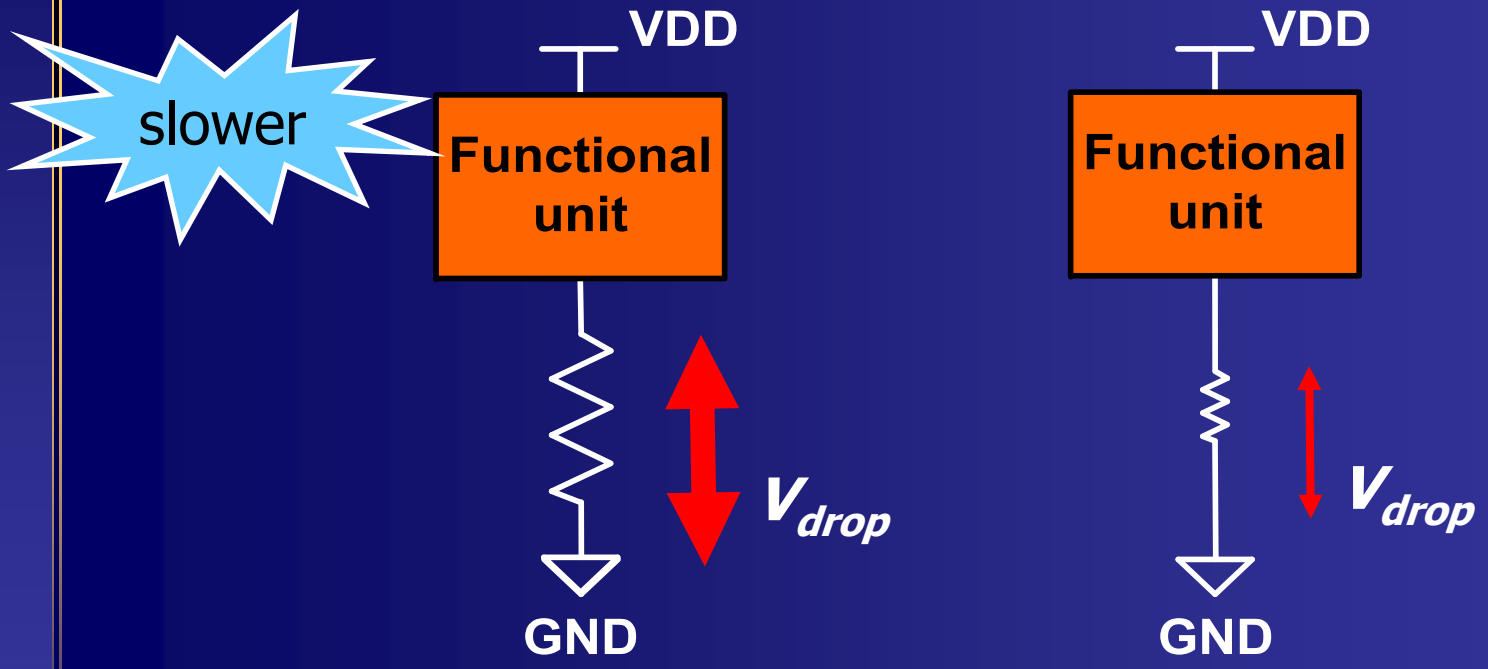
Power Gating Technique (1/2)

- In standby mode : (the sleep transistor is turned off)
  - The standby leakage current of the functional unit is proportional to the size of the sleep transistor.
  - Small sleep transistor to reduce leakage



Power Gating Technique (2/2)

- In active mode : (the sleep transistor is turned on and works as a resistor)
  - The sleep transistor produces a voltage drop that degrades the speed of the functional unit.



Pervious Works

- Up to now, the impact of high-level synthesis on the maximum allowable delays of functional units (for a target clock period) has not been studied.
  - Since the clock skew is assumed to be zero, the maximum allowable delay of each functional unit is definitely the target clock period; thus, there is no need to study this problem.
- However, in modern high-speed circuit design, the clock skew is often intentionally utilized to improve the circuit performance.

Our Contributions

- In this paper, we present the first work to formally draw up the timing driven power gating problem in the high-level synthesis of non-zero clock skew circuits.
  - Given a target clock period and design constraints, our objective is to derive the minimum-standby-leakage-current resource binding solution.
- Our work includes the following two aspects
  - First, we propose an MILP (mixed integer linear programming) approach to guarantee obtaining the optimal solution.
  - Second, we also propose a heuristic approach to deal with the same problem in polynomial time complexity.

Outline

- Introduction
- Motivation
- Our Approach
  - MILP
  - Heuristic Algorithm
- Experimental Result
- Conclusion

Functional Units with Power Gating

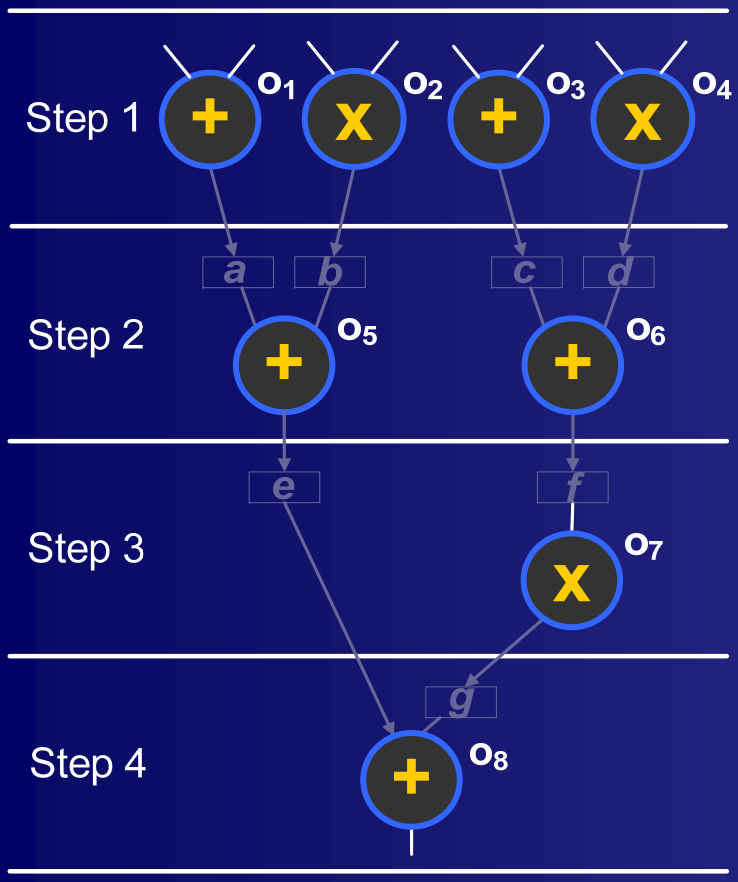
Suppose we are given two multipliers, called mul_1 and mul_2, and two adders, called add_1 and add_2:

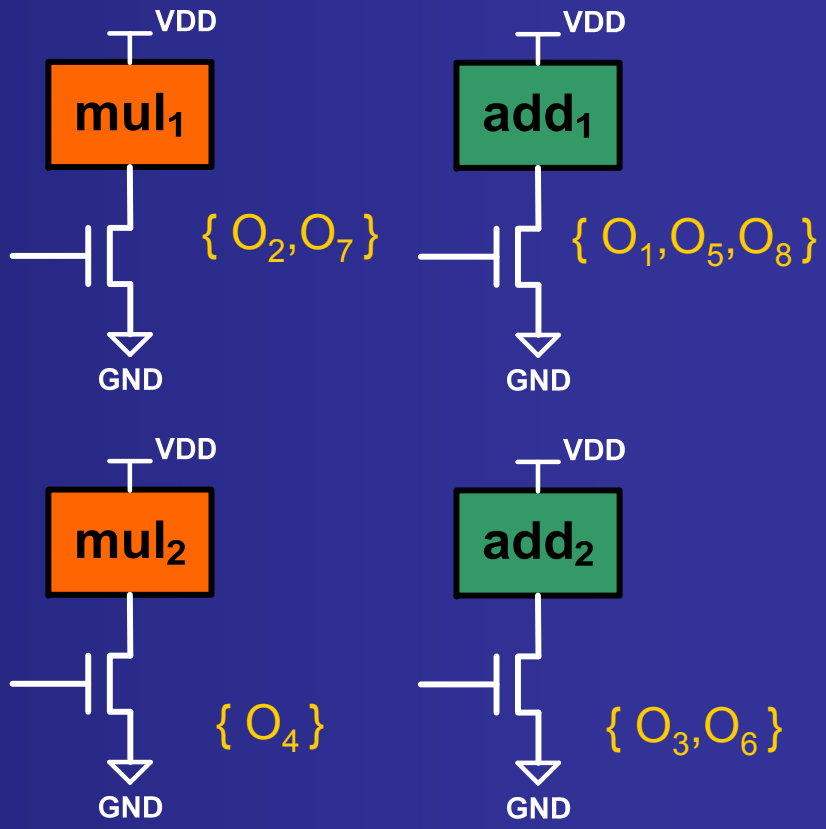
Type of Functional Unit	Transistor Size	Delay (min,max)	Leakage
Multiplier (mul)	Small (S)	(34,40)	5
	Medium (M)	(33,38)	20
	Large (L)	**(28,34)**	35
Adder (add)	Small (S)	(10,12)	4
	Large (L)	**(8,10)**	5

If the power gating implementation selection is
mul_1(fast), mul_2(fast), add_1(fast), and add_2(fast)
Total standby leakage current is 80 (due to 35+35+5+5)

Resource Binding (Functional Unit)

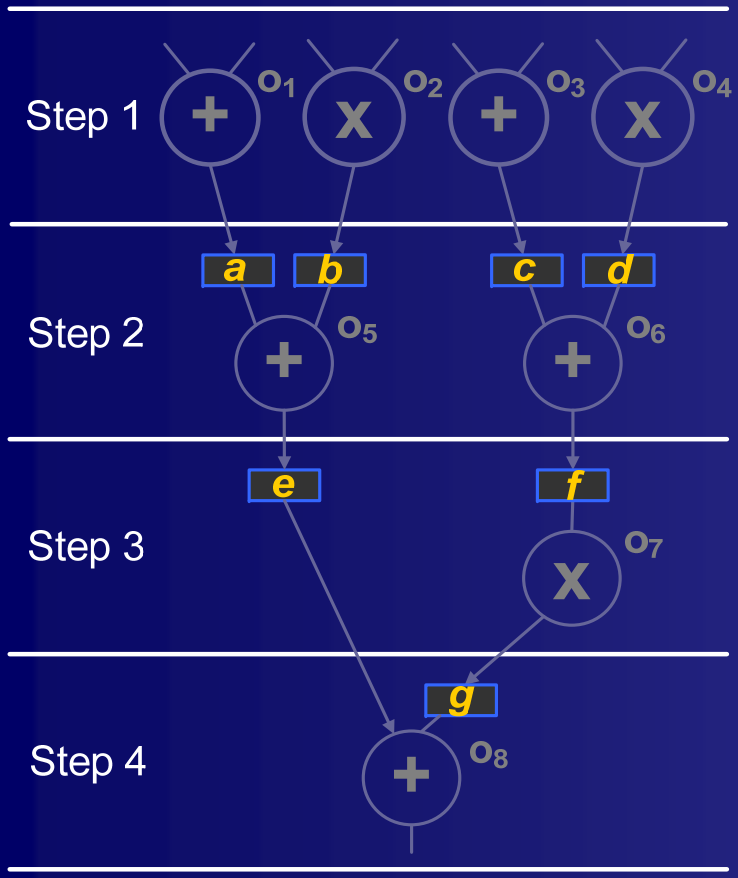
- Suppose we are given two multipliers, called mul_1 and mul_2, and two adders, called add_1 and add_2:

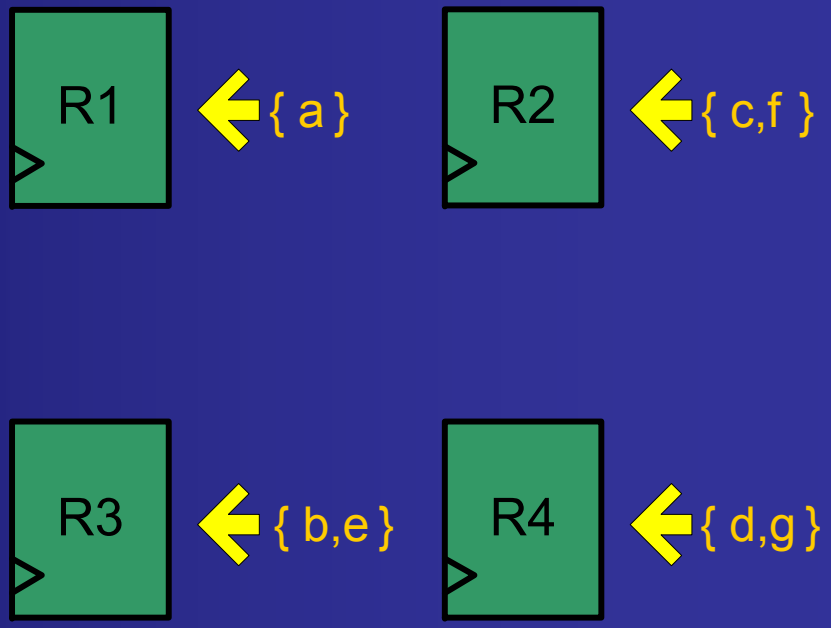




Resource Binding (Register)

- Suppose we are given four registers, called R1, R2, R3, and R4:

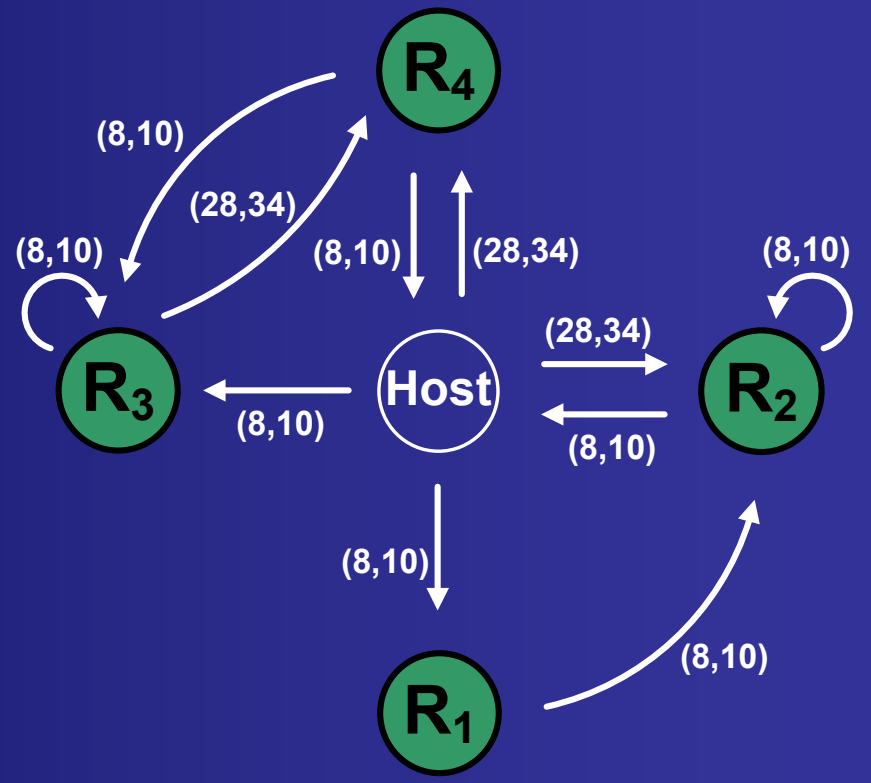




Circuit Graph of Resource Binding

$mul_1(fast) = \{o_2, o_7\}$,
$mul_2(fast) = \{o_4\}$,
$add_1(fast) = \{o_1, o_5, o_8\}$,
$add_2(fast) = \{o_3, o_6\}$,
R1 = {a}, R2 = {b,e},
R3 = {c,f}, and R4 = {d,g}.

Resource Binding



Circuit Graph

If the clock skew is zero,
the clock period cannot be less than 34.

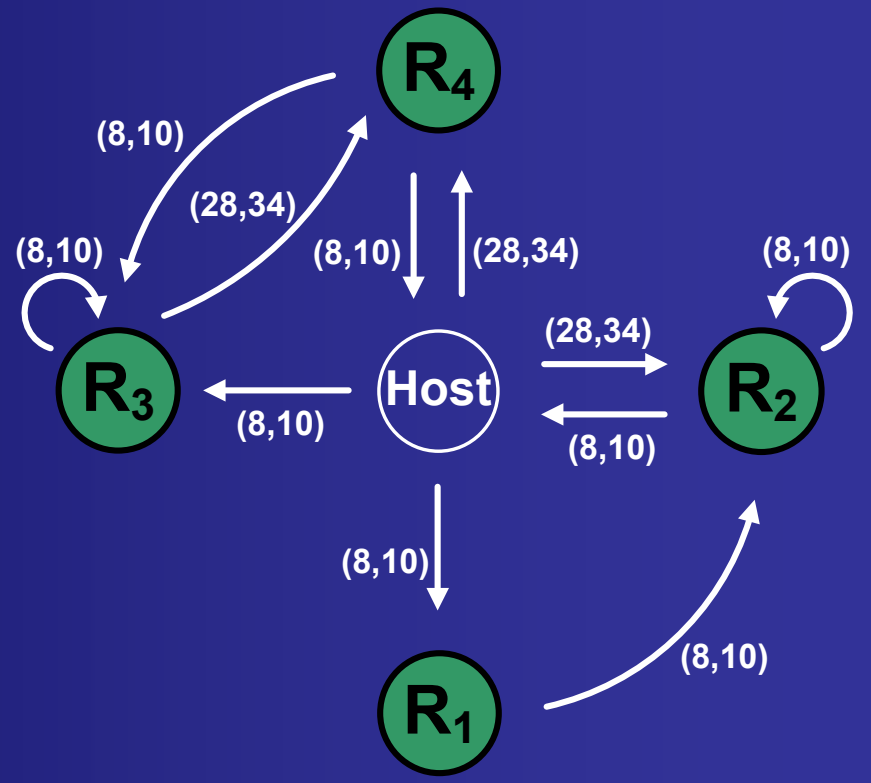
Min-Period Clock Skew Scheduling

- By properly scheduling the clock arrival time of registers, the clock period can be shorter than the longest combinational delay.
- Several graph-based algorithms use the constraint graph to solve the optimal clock skew scheduling.

DRAWBACK OF EXISTING FLOW

mul_1(fast) = {o_2,o_7},
mul_2(fast) = {o_4},
add_1(fast) = {o_1,o_5,o_8},
add_2(fast) = {o_3,o_6},
R1 = {a}, R2 = {b,e},
R3 = {c,f}, and R4 = {d,g}.

Resource Binding



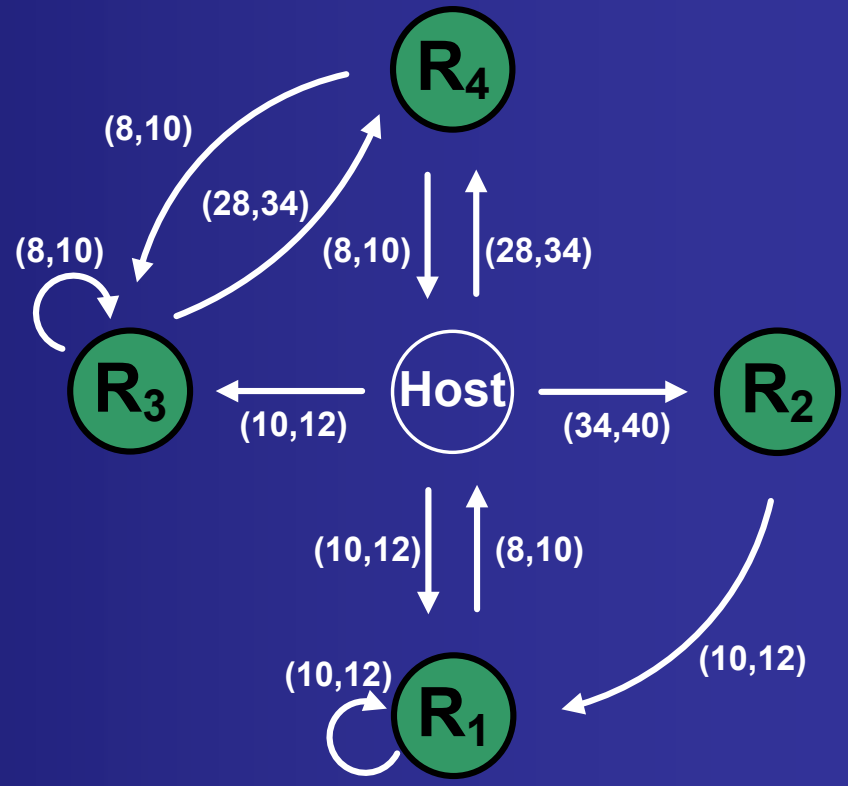
Circuit Graph

The smallest feasible clock period is 22.
Total standby leakage current is 80.

Our Solution

mul_1(fast) = {o_4,o_7},
mul_2(slow) = {o_2},
add_1(fast) = {o_3,o_6,o_8},
add_2(slow) = {o_1,o_5},
R1 = {a,e}, R2 = {b},
R3 = {c,f}, and R4 = {d,g}.

Resource Binding



Circuit Graph

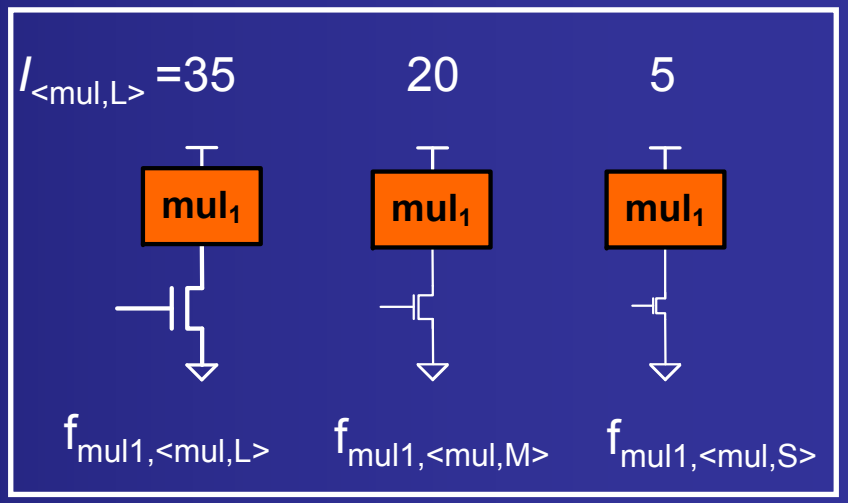
The smallest feasible clock period is only 22.
Total standby leakage current is only 49.

Outline

- Introduction
- Motivation
- Our Approach
  - MILP
  - Heuristic Algorithm
- Experimental Result
- Conclusion

Objective Function

- Our objective function is to minimize total standby leakage current.



- Minimize

$$\sum_{z \in Q} \sum_{w \in h(e(z))} f_{z,<e(z),w>} \cdot I_{<e(z),w>}.$$

Minimize

$f_{mul1,<mul,L>} \times 35 + f_{mul1,<mul,M>} \times 20 + f_{mul1,<mul,S>} \times 5 +$
$f_{mul2,<mul,L>} \times 35 + f_{mul2,<mul,M>} \times 20 + f_{mul2,<mul,S>} \times 5 +$
$f_{add1,<add,L>} \times 5 + f_{add1,<add,S>} \times 4 + f_{add2,<add,L>} \times 5 +$
$f_{add2,<add,S>} \times 4.$

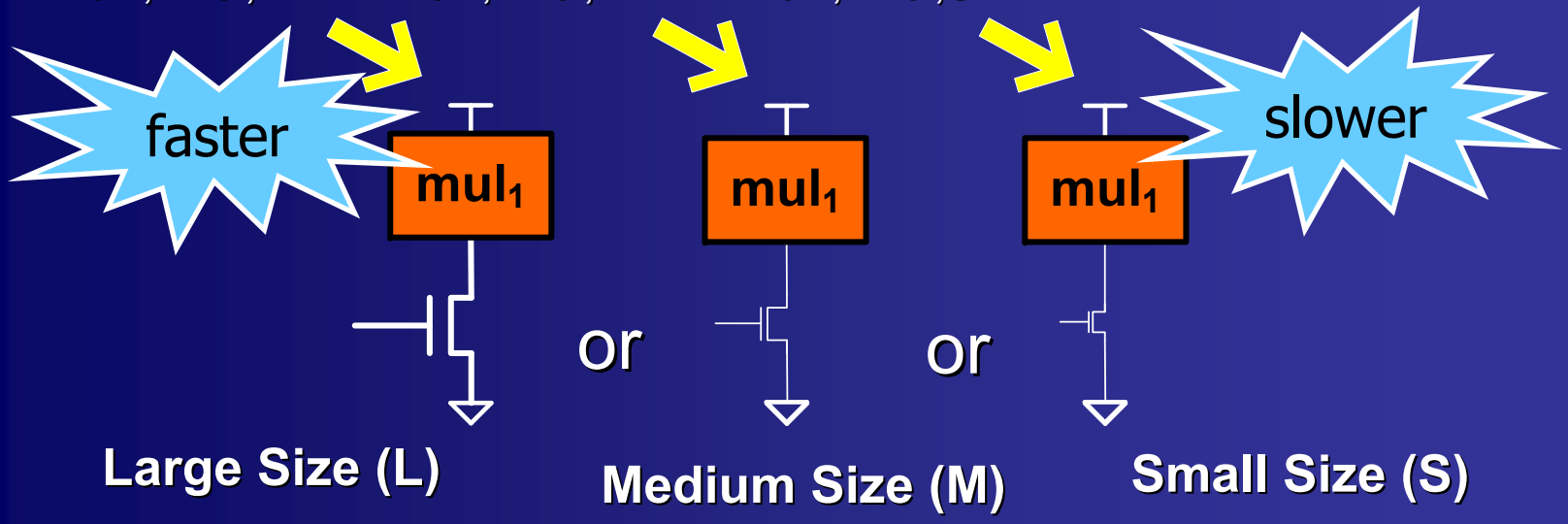
Formula 2

- Each functional unit must select one implementation. Thus, for each functional unit z, we have the following constraint:

$$\sum_{w \in h(e(z))} f_{z,<e(z),w>} = 1.$$

- In this example, we have the following constraints:

$f_{mul1,<mul,L>} + f_{mul1,<mul,M>} + f_{mul1,<mul,S>} = 1$; and so on.

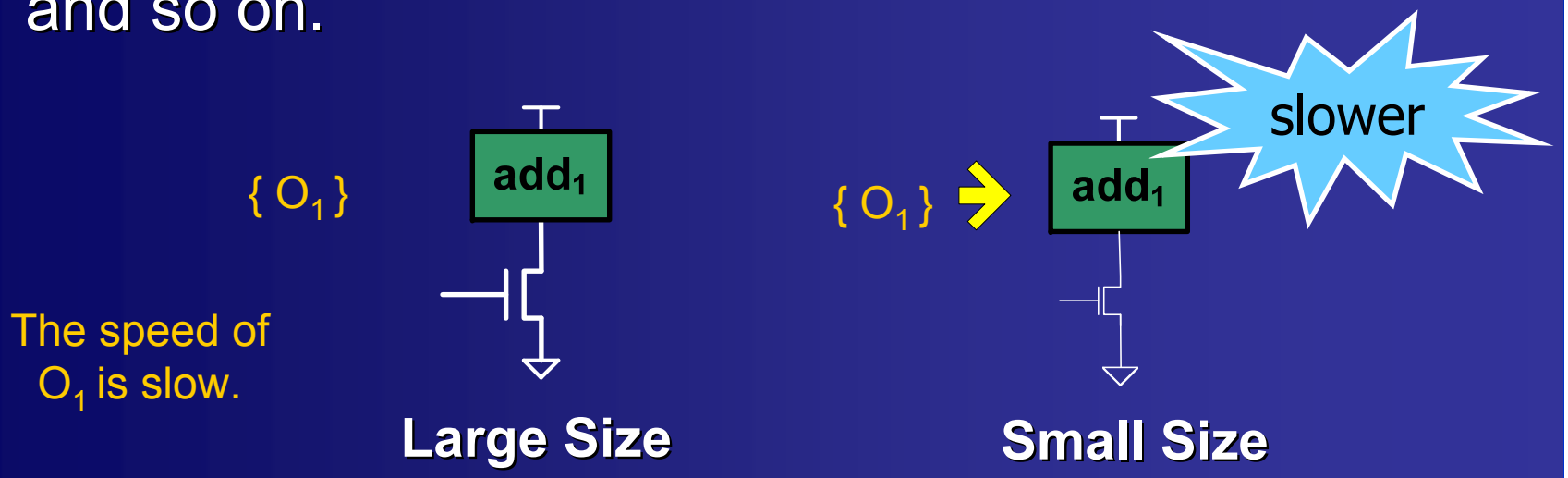


Formula 3

- If the functional unit z is not implemented by <e(z),w>, then the value of binary variable $y_{j,z,<e(z),w>}$ is definitely to be 0. Therefore, we have the following constraint: $y_{j,z,<e(z),w>} \leq f_{z,<e(z),w>}$.
- In this example, we have the following constraints:

  $y_{o1,add1,<add,L>} \leq f_{add1,<add,L>}$;

  and so on.

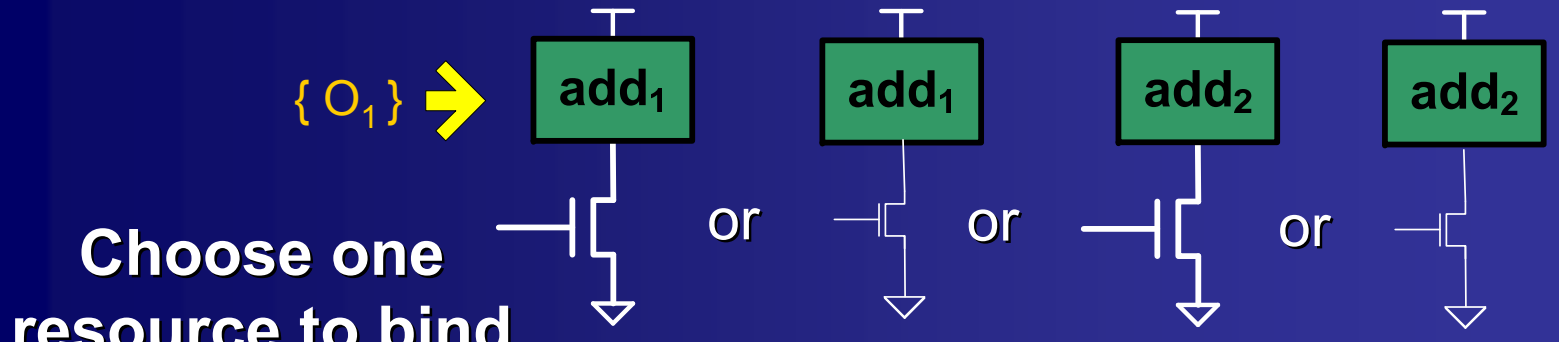


Formula 4

- Each operation must be assigned to one functional unit. Therefore, for each operation o_j, we have the following constraint:

$$\sum_{z \in c(g(j))} \sum_{w \in h(g(j))} y_{j,z,<e(z),w>} = 1.$$

- In this example, we have the following constraints: $y_{o1,add1,<add,L>} + y_{o1,add1,<add,S>} + y_{o1,add2,<add,L>} + y_{o1,add2,<add,S>} = 1$; and so on.



Formula 5

- If two operations have overlapping lifetimes, they cannot share the same functional unit. Thus, we have the following constraint:

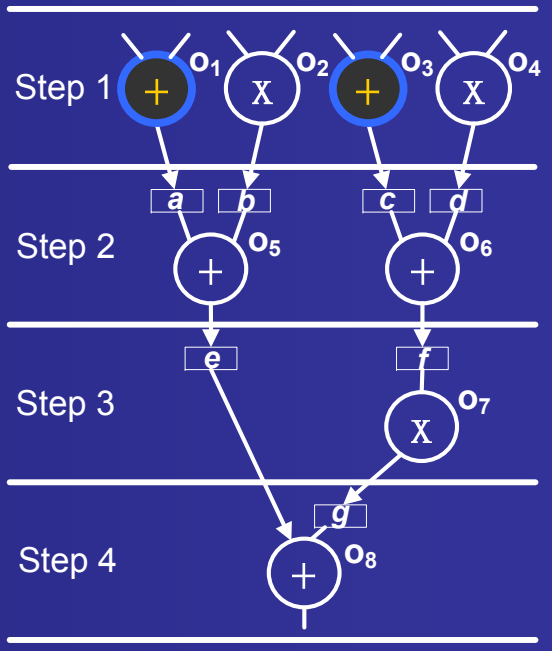
  $y_{j,z,<e(z),w>} + y_{k,z,<e(z),w>} \leq 1.$

- In this example, we have the following constraints:

  $y_{o1,add1,<add,L>} + y_{o3,add1,<add,L>} \leq 1;$

  $y_{o1,add1,<add,S>} + y_{o3,add1,<add,S>} \leq 1;$

  and so on.



O_1, O_3 Life Time conflict

Formula 6

- Let P be a constant that denotes the target clock period. For the input variable u and the output variable v of operation o_j, the maximum allowable delay must satisfy the setup constraint:

$$\sum_{z \in c(g(j))} \sum_{w \in h(g(j))} y_{j,z,<e(z),w>} \cdot D_{<e(z),w>} \leq P - T_u + T_v.$$

- In this example, we have the following constraints:

$y_{o1,add1,<add,L>} \times 10 + y_{o1,add1,<add,S>} \times 12 +$
$y_{o1,add2,<add,L>} \times 10 + y_{o1,add2,<add,S>} \times 12$
$\leq P - T_{host} + T_a$; (P = 22)
and so on.

Formula 7

- For the input variable u and the output variable v of operation o_j, the minimum allowable delay must satisfy the hold constraint:

$$T_v - T_u \leq \sum_{z \in c(g(j))} \sum_{w \in h(g(j))} y_{j,z,<e(z),w>} \cdot d_{<e(z),w>}.$$

- In this example, we have the following constraints:

  $T_a - T_{host} \leq y_{o1,add1,<add,L>} \times 8 + y_{o1,add1,<add,S>} \times 10 +$
  $y_{o1,add2,<add,L>} \times 8 + y_{o1,add2,<add,S>} \times 10;$
  and so on.

- Formula 8~11 (Use the Register Binding Approach in [7])

Outline

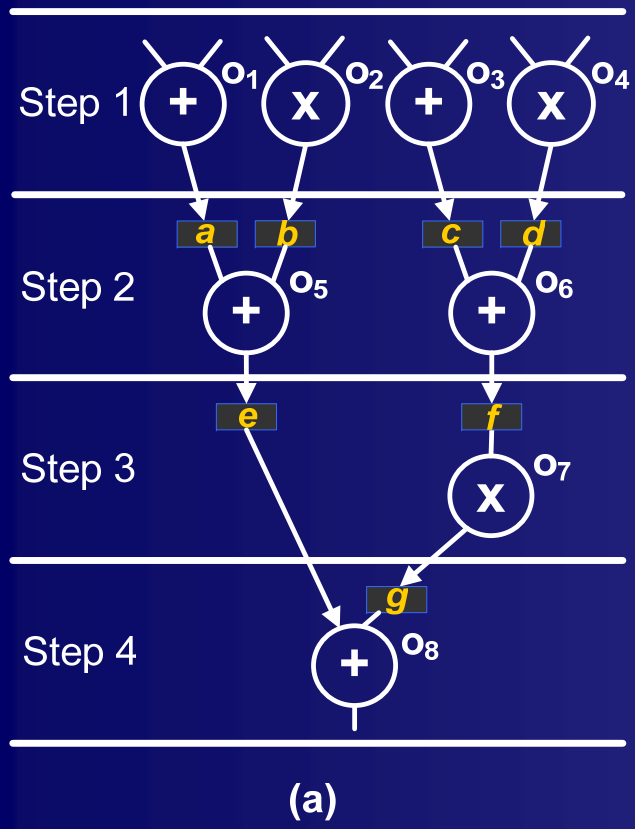
- Introduction
- Motivation
- Our Approach
  - MILP
  - Heuristic Algorithm
- Experimental Result
- Conclusion

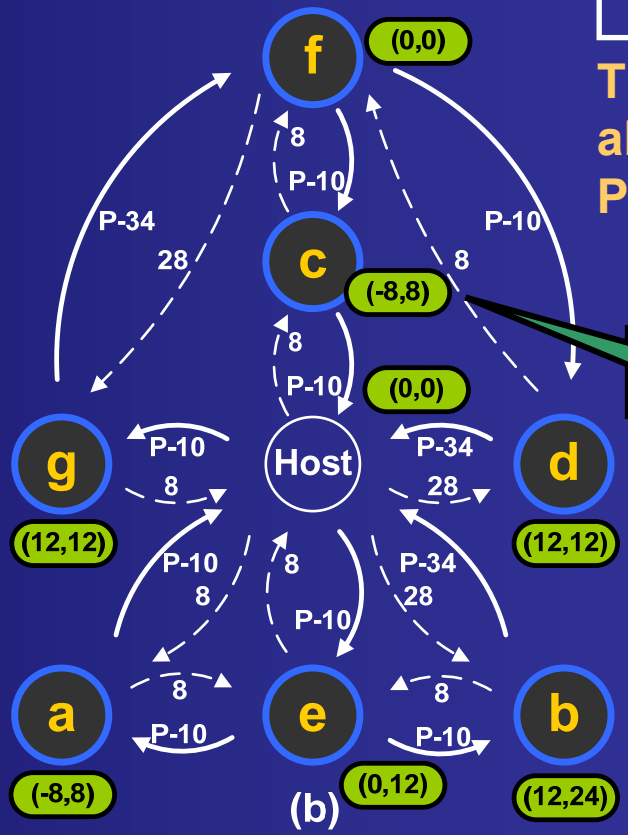
Heuristic Approach

- We provide a method to estimate the maximum allowable delay of each operation before the resource binding.
- We try to assign the operations that have similar maximum allowable delay to the same functional unit.
- As a result, we can have more non-critical functional units.

Step 1: Estimate the Maximum Allowable Delay

- The estimated maximum allowable delays of operations o_1, o_2, o_3, o_4, o_5, o_6, o_7, and o_8 are 30, 46, 30, 34, 22, 10, 34, and 10, respectively.





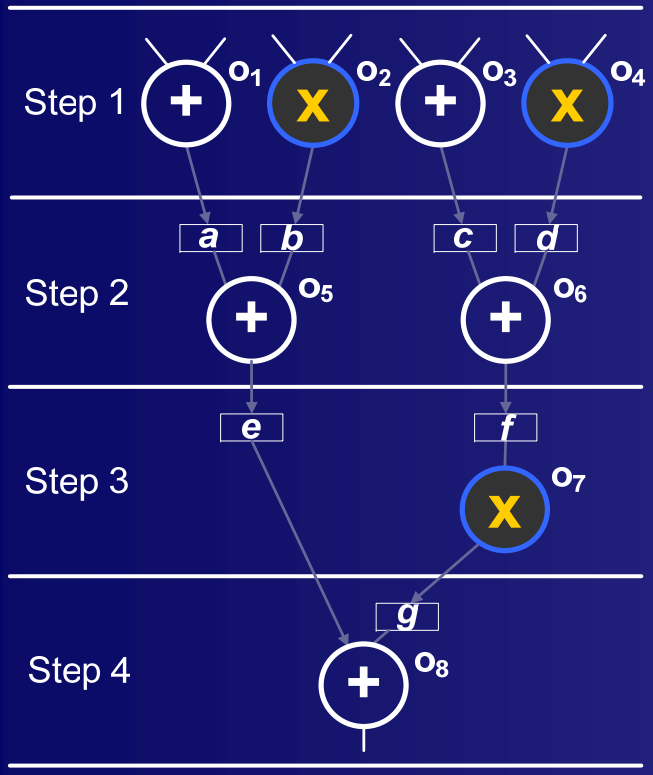
P = 12

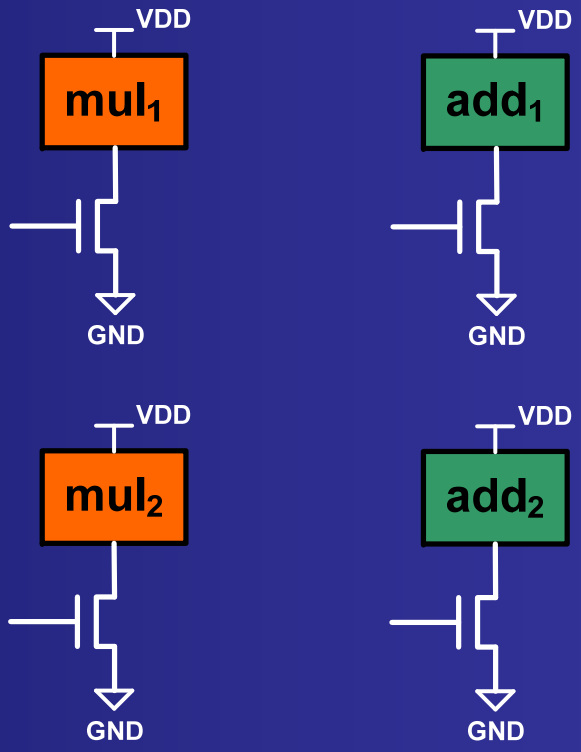
The maximum allowable delays $P-E_u+L_v$

E_c =-8, L_c =8

Step 2: Power Gating Implementation (1/5)

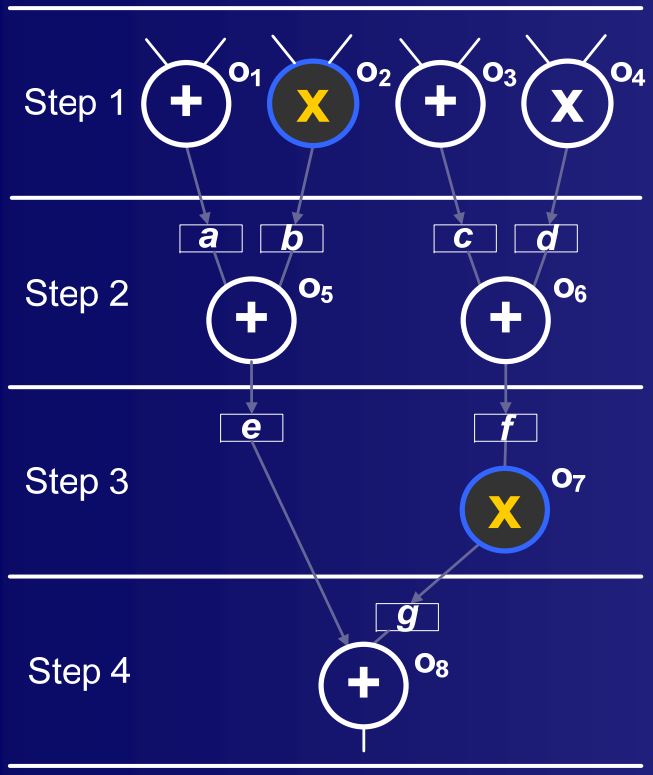
- We consider the multiplier type. The estimated maximum allowable delays of operations o_2, o_4, and o_7 are 46, 34, and 34, respectively. (o_4, o_7 high priority)

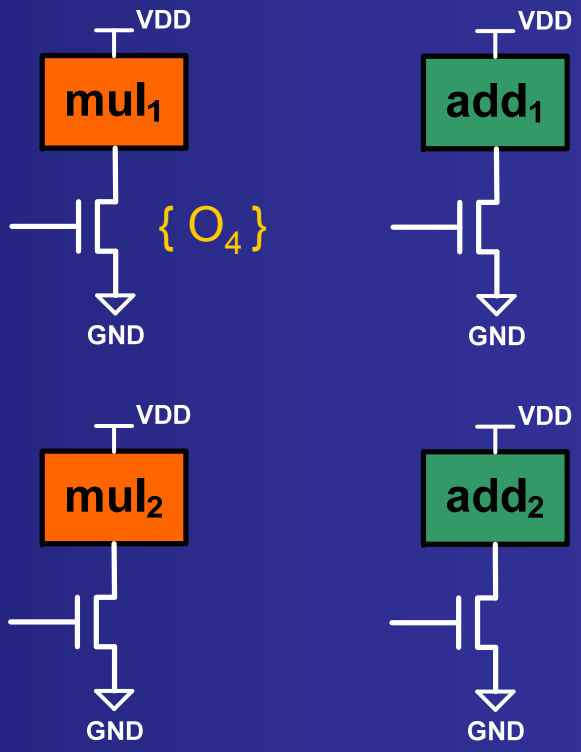




Step 2: Power Gating Implementation (2/5)

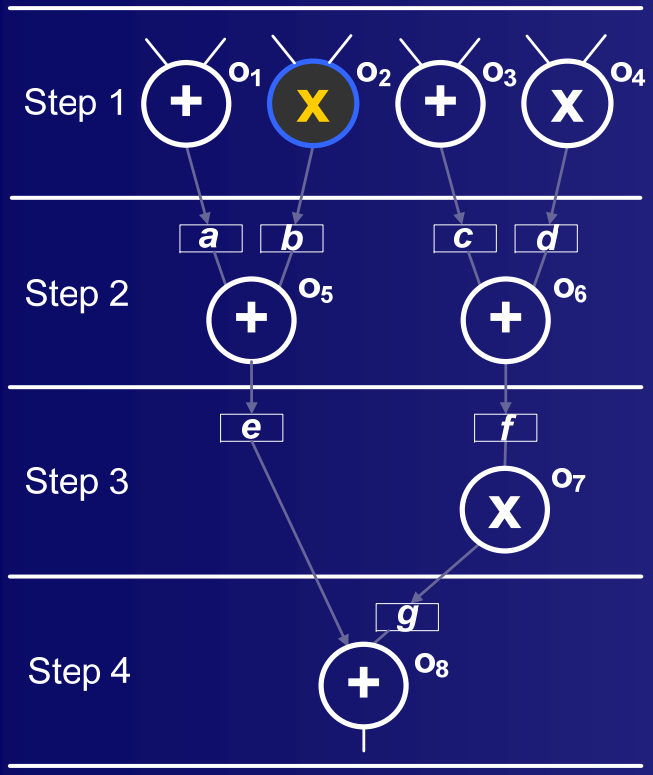
- Suppose operation o_4 is chosen. We assign operation o_4 to mul_1(fast). The estimated maximum allowable delays of operation o_7 is 34.

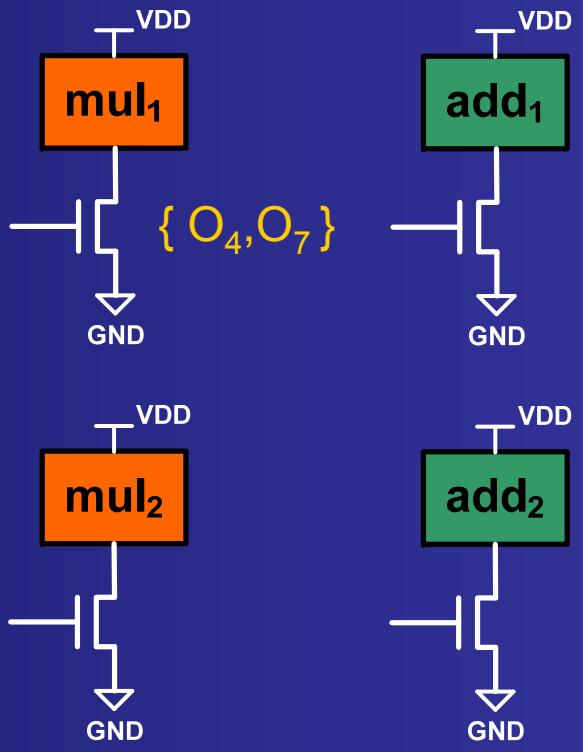




Step 2: Power Gating Implementation (3/5)

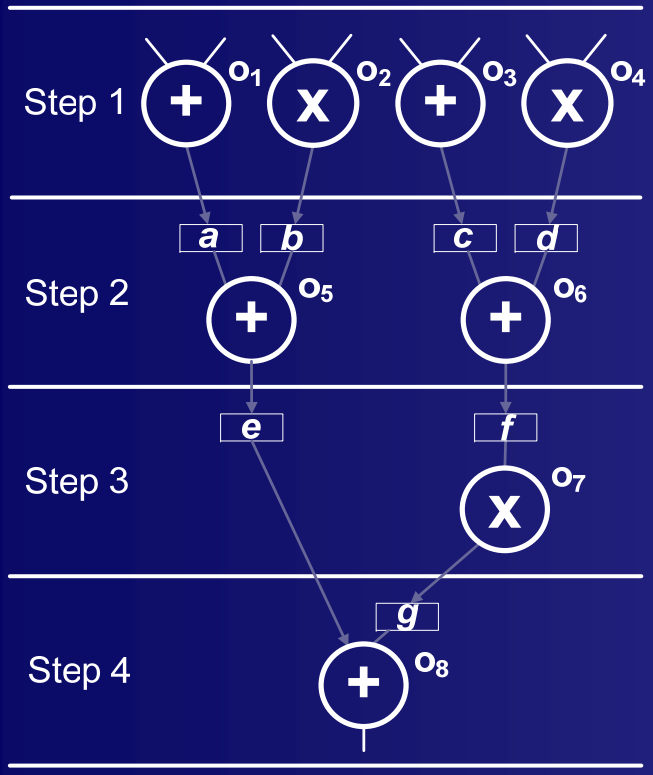
- We assign operation o_7 to mul_1(fast). The estimated maximum allowable delays of operation $\mathbf{o_7}$ is **34**.

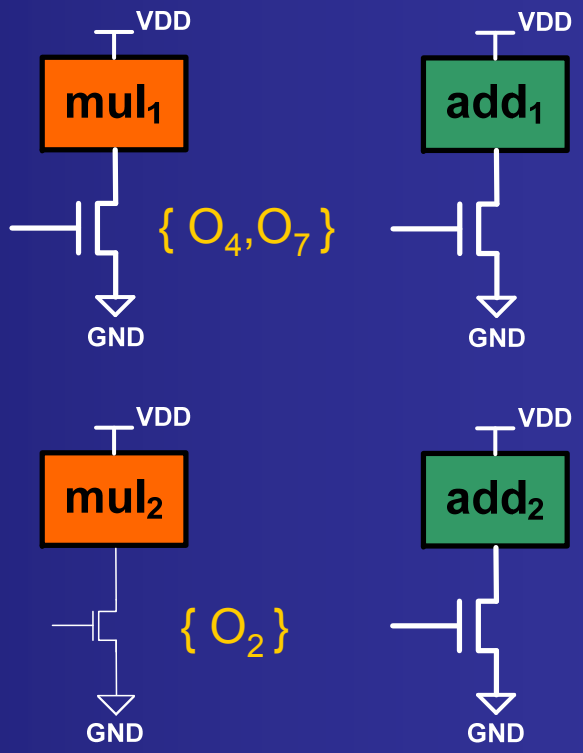




Step 2: Power Gating Implementation (4/5)

- Because of the lifetime constraint, we find operation o_2 cannot be assigned to the functional unit mul_1. We assign operation o_2 to mul_2(slow).

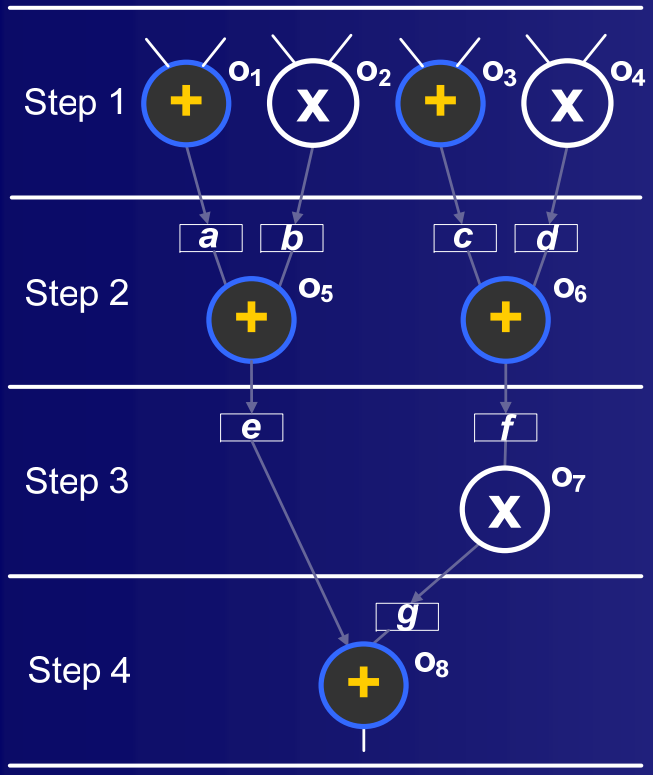


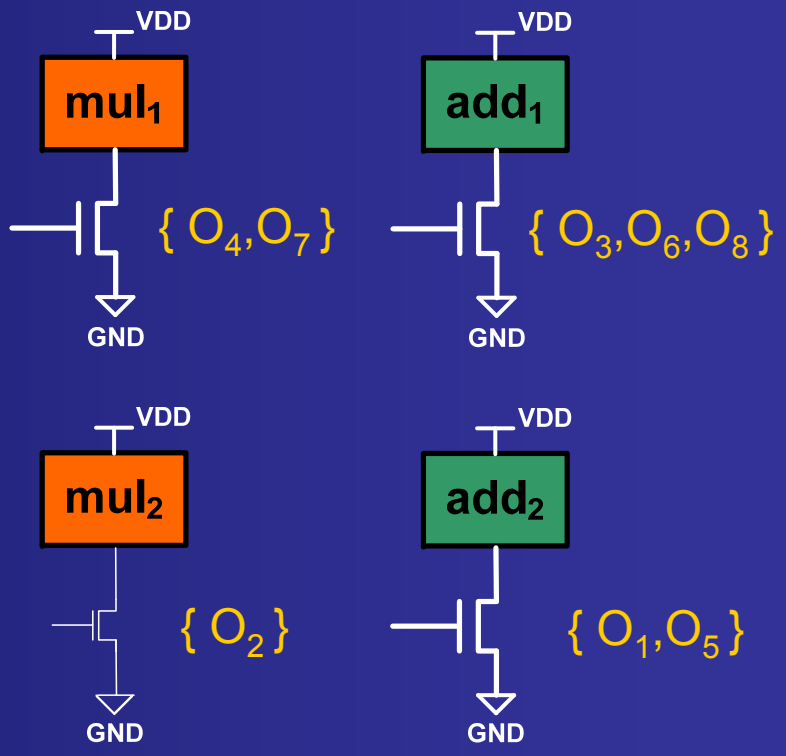


o_2 maximum allowable delays = **46**.

Step 2: Power Gating Implementation (5/5)

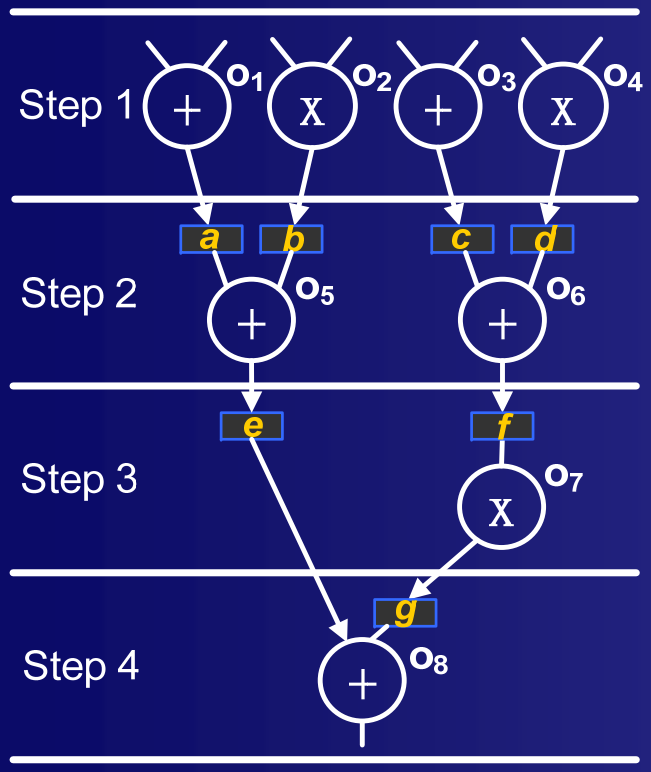
- Similarly, we consider the adder type. We have $add_1(fast) = \{o_3, o_6, o_8\}$, $add_2(slow) = \{o_1, o_5\}$.

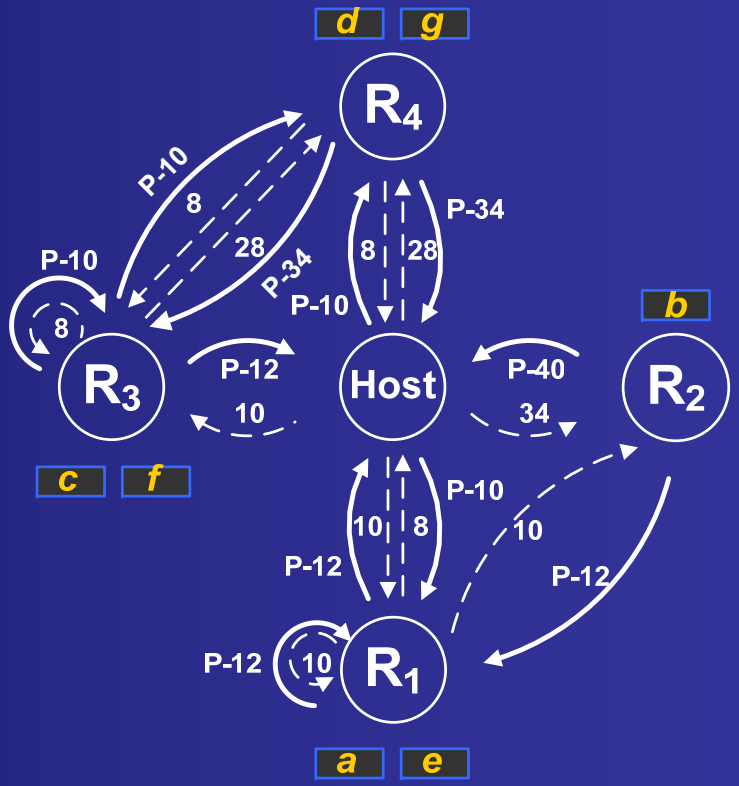




Step 3:
Use the Register Binding Approach in [7]

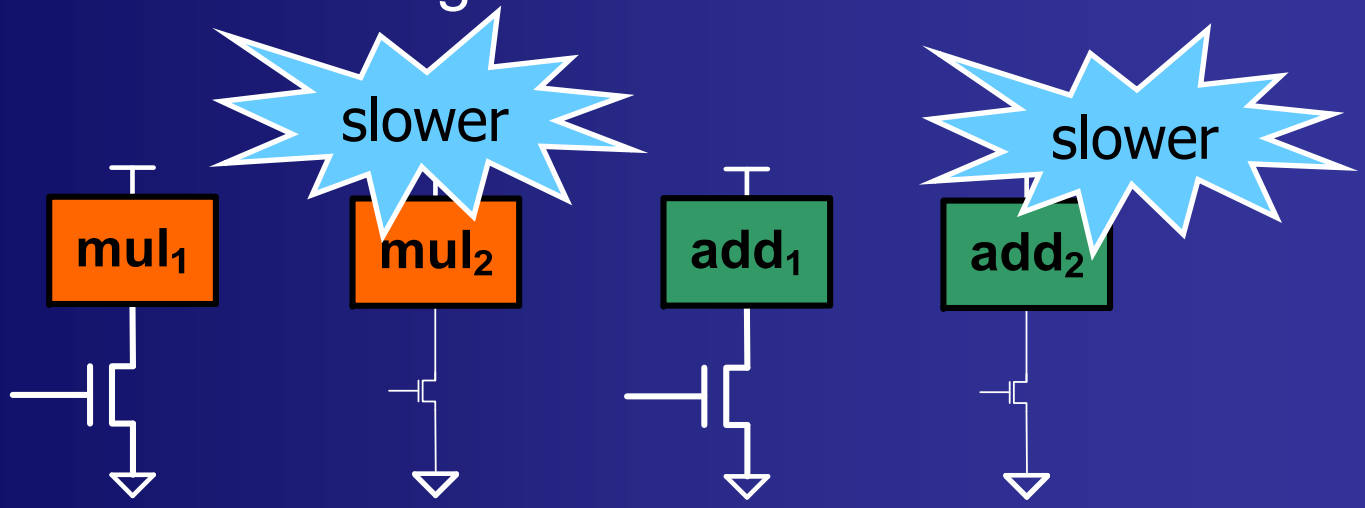
- We use [7] to perform register binding for clock period minimization. we have R1 = {a,e}, R2 = {b}, R3 = {c,f}, and R4 = {d,g}. The smallest feasible clock period is 22.





Step 4:
Adjust the Power Gating Implementation

- Now, the actual maximum allowable delay of each functional unit can be calculated based on the target clock period and the clock skew schedule derived in the third step.
- We adjust the power gating implementation of each functional unit according to its actual maximum allowable delay.



The total standby leakage current is only 49.

Outline

- Introduction
- Motivation
- Our Approach
  - MILP
  - Heuristic Algorithm
- Experimental Result
- Conclusion

Experimental Environment

- Our platform is Windows XP operating system running on AMD K8-4200+ processor.
  - We use Extended-LINGO Release 10.0 as the mixed integer linear program solver and use C programming language to implement the process of solution space reduction.
- We compare our approach with existing design flow.

Circuit				Improvement	
	Existing Flow	Our MILP	Our Heuristic	Our MILP	Our Heuristic
HAL	1.764	0.484	0.484	72.56%	72.56%
AR	2.644	0.940	1.152	64.45%	56.43%
BF	1.336	0.484	0.484	63.77%	63.77%
EWF	1.350	0.498	0.738	63.11%	45.33%
IDCT1	3.585	1.458	1.956	59.33%	45.44%
Motion	7.376	2.607	4.118		

64.65%

54.62%

Concluding Remarks

- This paper presents the first work to formally formulate the timing driven power gating in the high-level synthesis of a non-zero clock skew circuit.
- Given a target clock period and design constraints, our goal is to find a resource binding solution so that the total standby leakage current is minimized.
- Compared with the existing design flow, our approach has a significant improvement without any overhead.

Thank you