Fast Multi-Domain Clock Skew Scheduling for Peak Current Reduction

Shih-Hsu Huang , Chia-Ming Chang,
and Yow-Tyng Nieh

Department of Electronic Engineering,
Chung Yuan Christian University, Taiwan

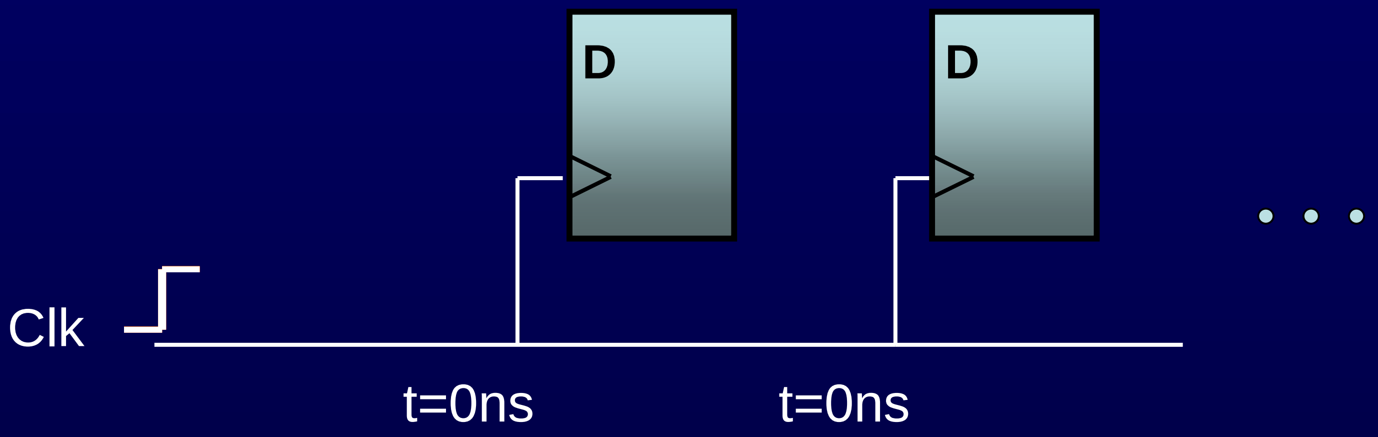
Outline

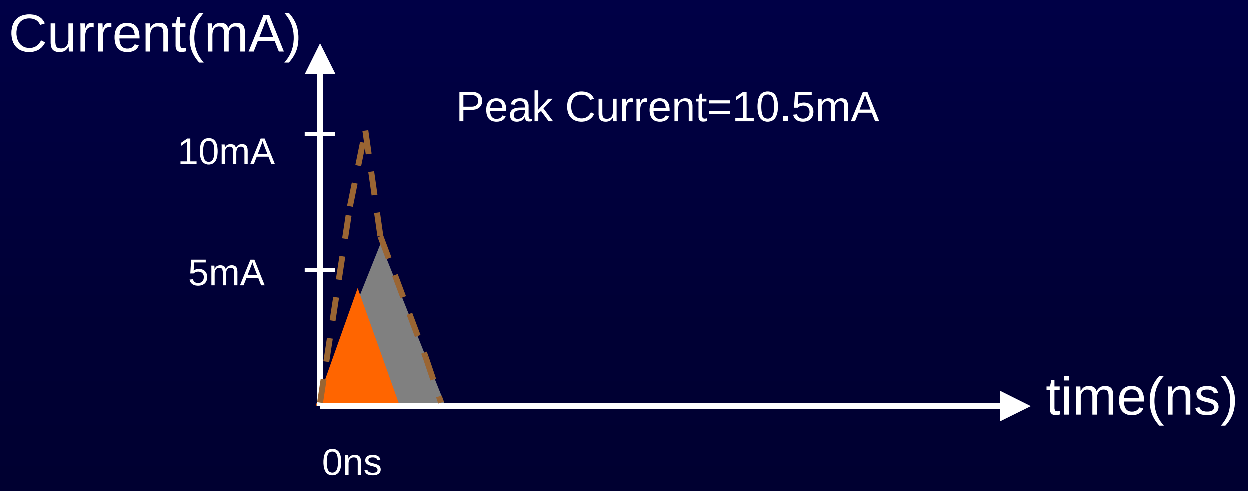
- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

Introduction

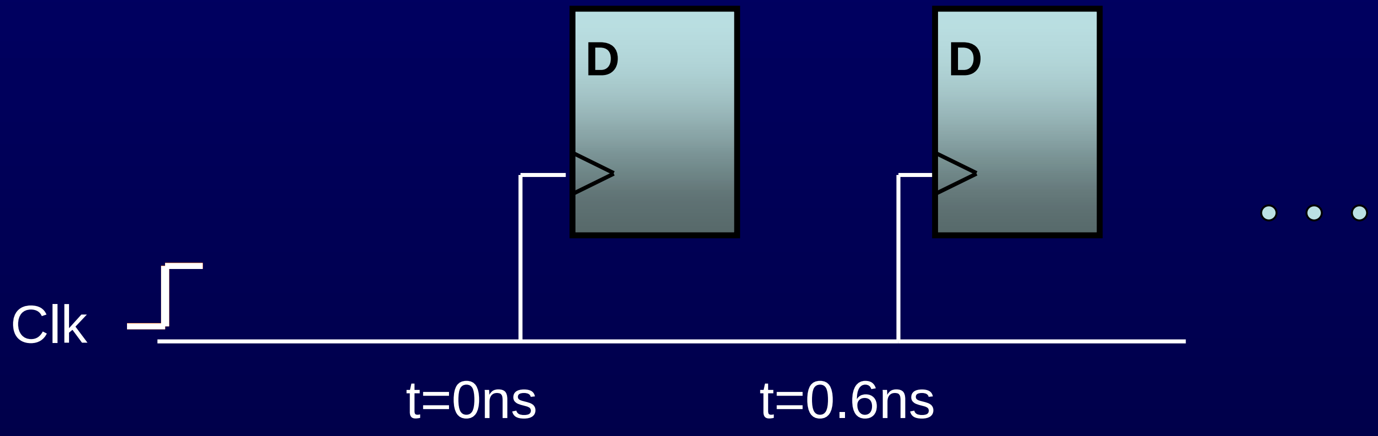
- In a high-speed synchronous digital circuit, a huge current peak is often observed at each clock edge due to the simultaneous signal switching.
- By skewing the clock inputs to various registers, it is possible to reduce this current peak.
- The peak current may lead to logic errors due to voltage drop or reliability problems due to electromigration.

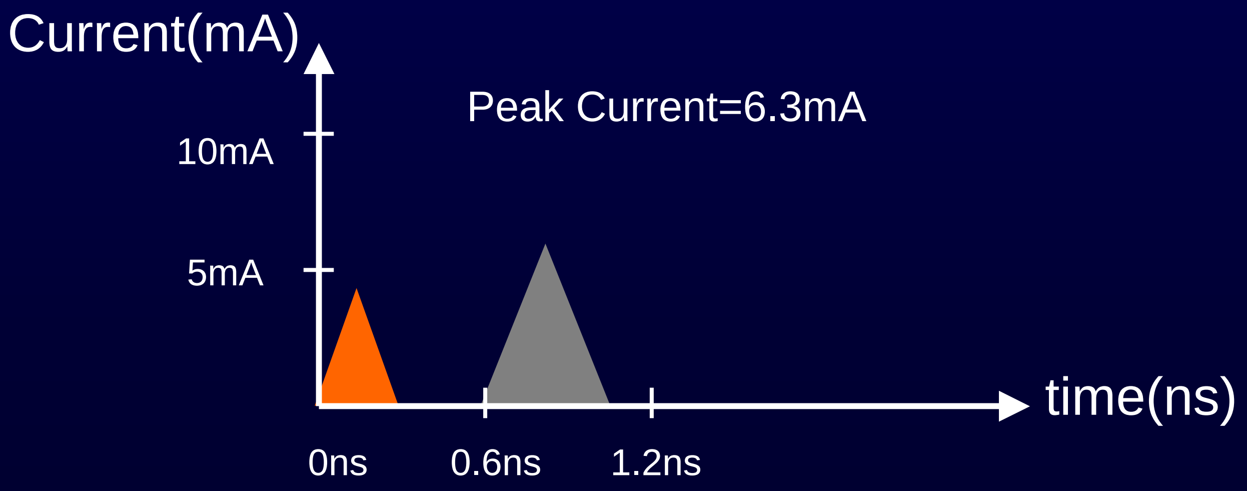
Zero Clock Skew Circuit





Non-Zero Clock Skew Circuit





Multiple Clocking Domains

- However, in fact, it is very difficult to implement a wide spectrum of dedicated clock delays.
- The architecture of multiple clocking domains, which restricts the clock arrival time of each register to specified clocking domains, provides an alternative to a wide spectrum of dedicated clock delays.

Previous Work

- Vittal, Ha, Brewer and Marek-Sadowska ever used the 0-1 integer linear programming (ILP) formulation to minimize the peak current under the constraint of multi-domain clock skew scheduling.
- Although their approach guarantees the optimality (in terms of the specified cost function), the run time of the 0-1 ILP solver grows dramatically with the increase of binary variables.
- The high computational expense has limited the use of their approach.

Our Approach

- We derive the ASAP (as-soon-as-possible) schedule and the ALAP (as-late-as-possible) schedule of each register. As a result, we can prune all the redundant binary variables and all the redundant constraints without sacrificing the exactness (optimality) of the solution.
- We also propose a zone-based scheduling algorithm to solve a large circuit heuristically. Experiments with large circuits consistently show that the zone-based scheduling achieves near-optimal solutions within an acceptable run time.

Outline

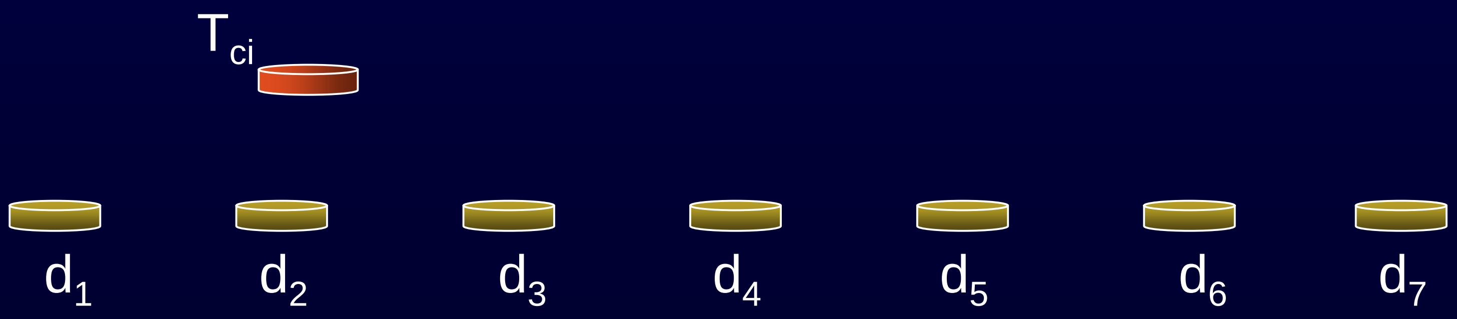
- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

Our Advantages

- The proposed algorithms, including ASAP scheduling, ALAP scheduling, and zone-based scheduling, are independent of the used peak current model (i.e., applicable to any peak current model).
- The peak current can be reduced without any penalty on clock cycle time and average power dissipation.

0-1 ILP Formulation

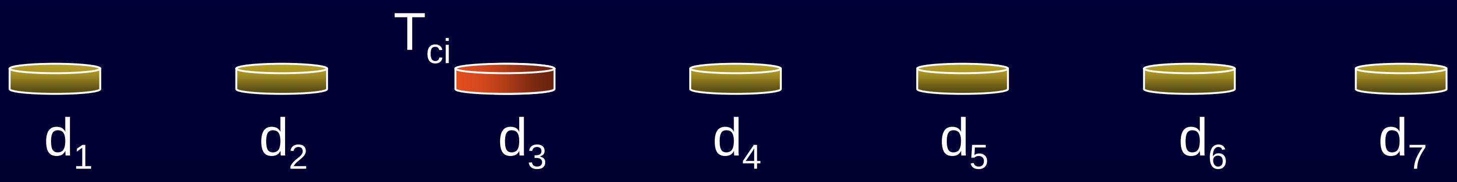
- Multi-domain clock skew scheduling is easier to implement because it only uses n discrete clocking domains: d_1, d_2, …and d_n
- The clock arrival time of register R_i, denoted as T_{Ci}, must be one of the n clocking domains.
- Binary variable $S_{i,k}$ is 1 if and only if T_{Ci} is equal to d_k.



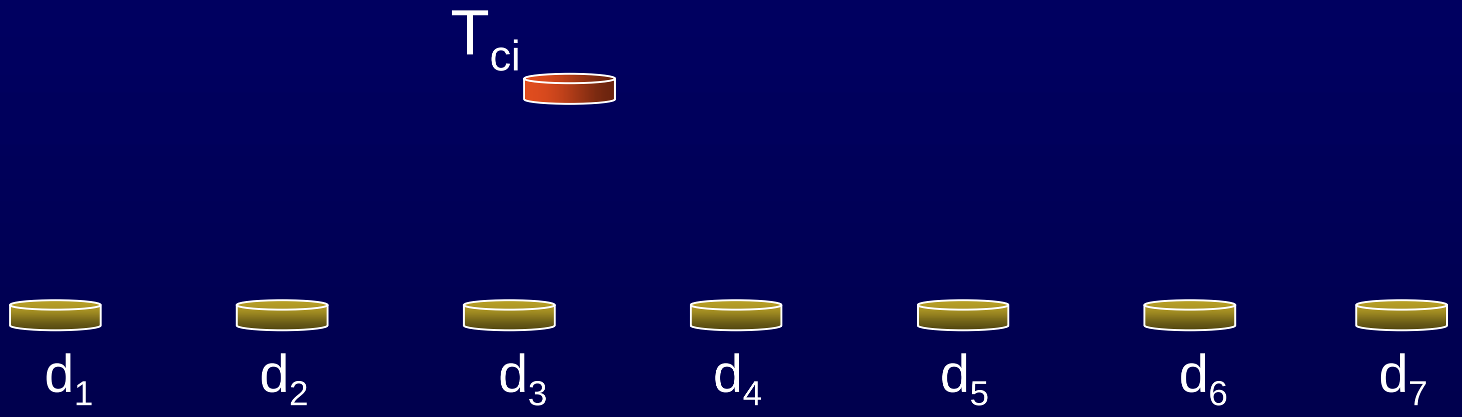
0-1 ILP Formulation

- Multi-domain clock skew scheduling is easier to implement because it only uses n discrete clocking domains: d_1, d_2, …and d_n
- The clock arrival time of register R_i, denoted as T_{Ci}, must be one of the n clocking domains.
- Binary variable $S_{i,k}$ is 1 if and only if T_{Ci} is equal to d_k.

$S_{i,3}=1$



Formulation 1



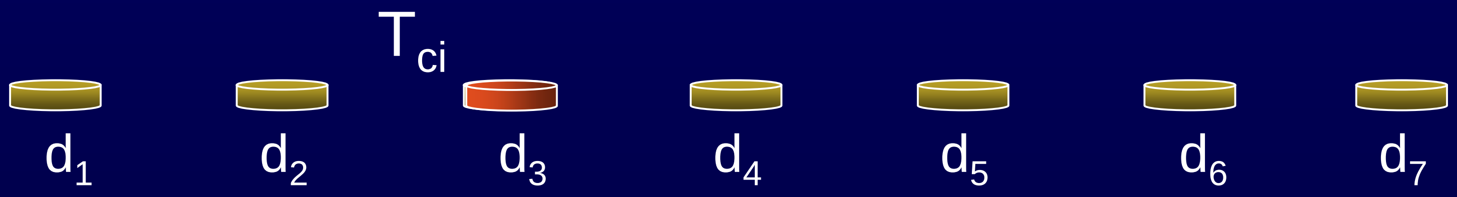
T_{ci}

d_1 d_2 d_3 d_4 d_5 d_6 d_7

$$S_{i,1} + S_{i,2} + S_{i,3} + S_{i,4} + S_{i,5} + S_{i,6} + S_{i,7} = 1$$

$$\sum_{k=1}^{n} S_{i,k} = 1.$$

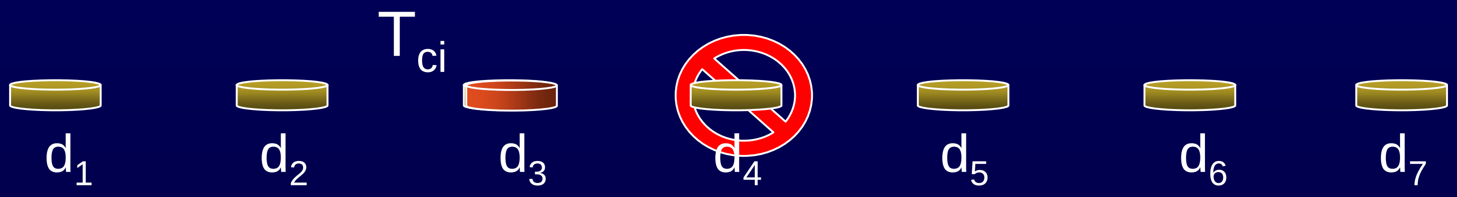
Formulation 1



$$S_{i,1} + S_{i,2} + S_{i,3} + S_{i,4} + S_{i,5} + S_{i,6} + S_{i,7} = 1$$

$$\sum_{k=1}^{n} S_{i,k} = 1.$$

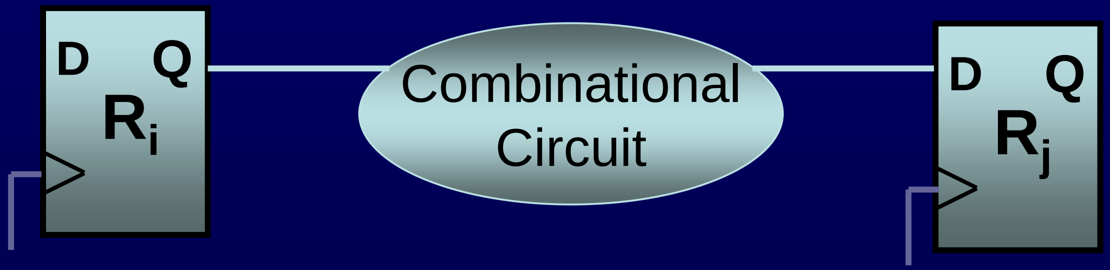
Formulation 1



$$S_{i,1} + S_{i,2} + S_{i,3} + S_{i,4} + S_{i,5} + S_{i,6} + S_{i,7} = 1$$

$$\sum_{k=1}^{n} S_{i,k} = 1.$$

Formulation 2



$T_{ci} \in \{d_1 \ldots d_n\}$

$T_{cj} \in \{d_1 \ldots d_n\}$

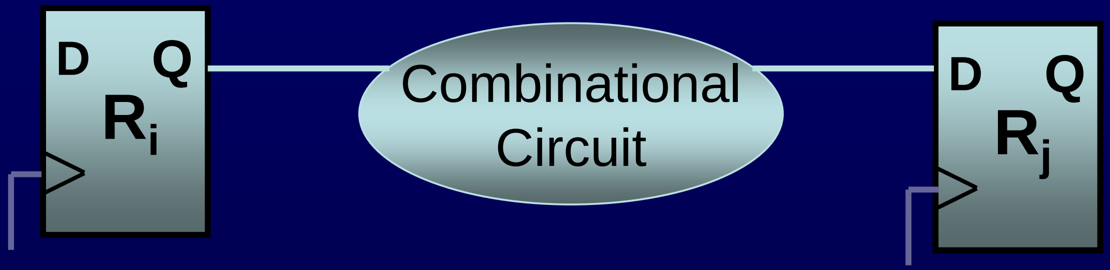
Double Clocking Constraint

$$T_{cj} - T_{ci} \le T_{PDi,\,j(\min)}$$

$$\Downarrow$$

$$\sum_{k=1}^{n} d_k \cdot S_{j,k} - \sum_{k=1}^{n} d_k \cdot S_{i,k} \le T_{PDi,\,j(\min)}$$

Formulation 3



$T_{ci} \in \{d_1 \ldots d_n\}$

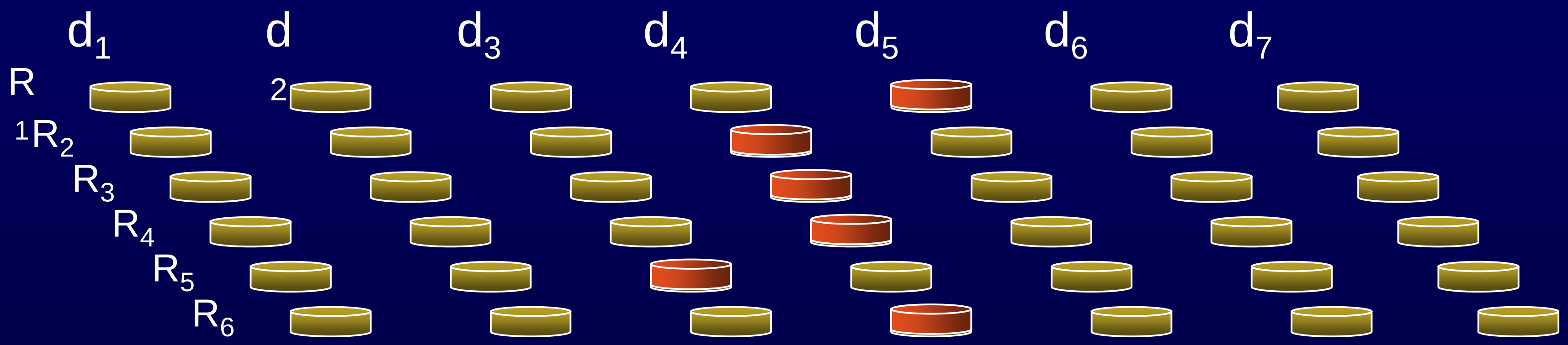
$T_{cj} \in \{d_1 \ldots d_n\}$

Zero Clocking Constraint

$$T_{ci} - T_{cj} \le P - T_{PDi,\,j(max)}$$

$$\sum_{k=1}^{n} d_k \cdot S_{i,k} - \sum_{k=1}^{n} d_k \cdot S_{j,k} \le P - T_{PDi,\,j(max)}$$

Formulation 4



Peak Current = 4

$$\sum_{i=1}^{r} S_{i,k} \leq \text{Peak\_Current}$$

The Objective is to minimize the value of Peak_Current (without loss of generality, here we assume the current of each register is the same).

Outline

- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

ASAP and ALAP

- Due to the zero clocking constraints and the double clocking constraints, many binary variables used in the 0-1 ILP formulations are definitely to be 0.
- We derive the ASAP schedule and the ALAP schedule to prune the redundancies without sacrificing the exactness (optimality) of the solution.
- For each register R_i, we use $ASAP_i$ to denote the earliest clock arrival time and $ALAP_i$ to denote the latest clock arrival time.

Forward Restriction

For each data path from register R_i to register R_j, if the range of T_{ci} is fixed, then the range of T_{cj} is restricted as below:

$$\mathrm{ASAP}_j \geq \mathrm{ASAP}_i + \mathrm{P} - \mathrm{T}_{\mathrm{PDi,\,j(max)}}$$

$$\mathrm{ALAP}_j \leq \mathrm{ALAP}_i + \mathrm{T}_{\mathrm{PDi,\,j(min)}}$$

Backward Restriction

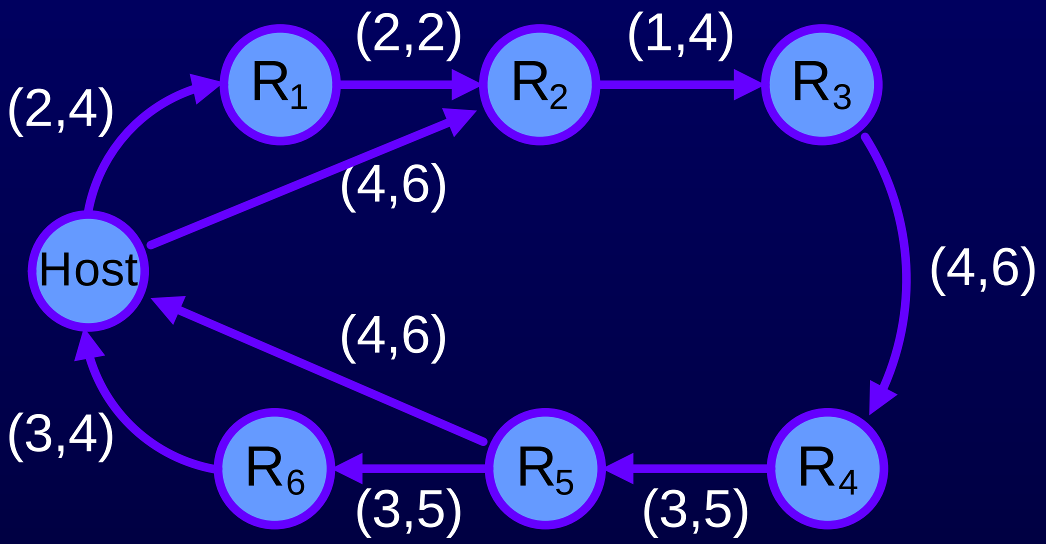
For each data path from register R_i to register R_j, if the range of T_{cj} is fixed, then the range of T_{ci} is restricted as below:

$$\mathrm{ASAP}_i \le \mathrm{ASAP}_j + \mathrm{P} - \mathrm{T}_{\mathrm{PDi,\,j(max)}}$$

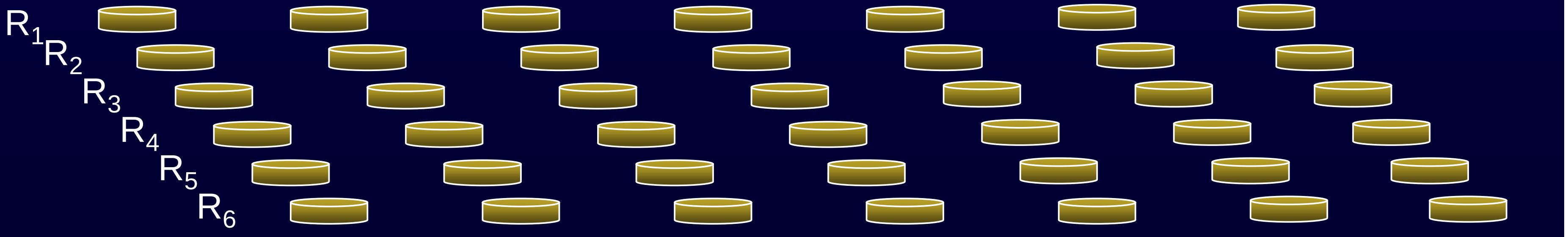
$$\mathrm{ALAP}_i \ge \mathrm{ALAP}_j - \mathrm{T}_{\mathrm{PDi,\,j(min)}}$$

An Example

Clock Period = 6

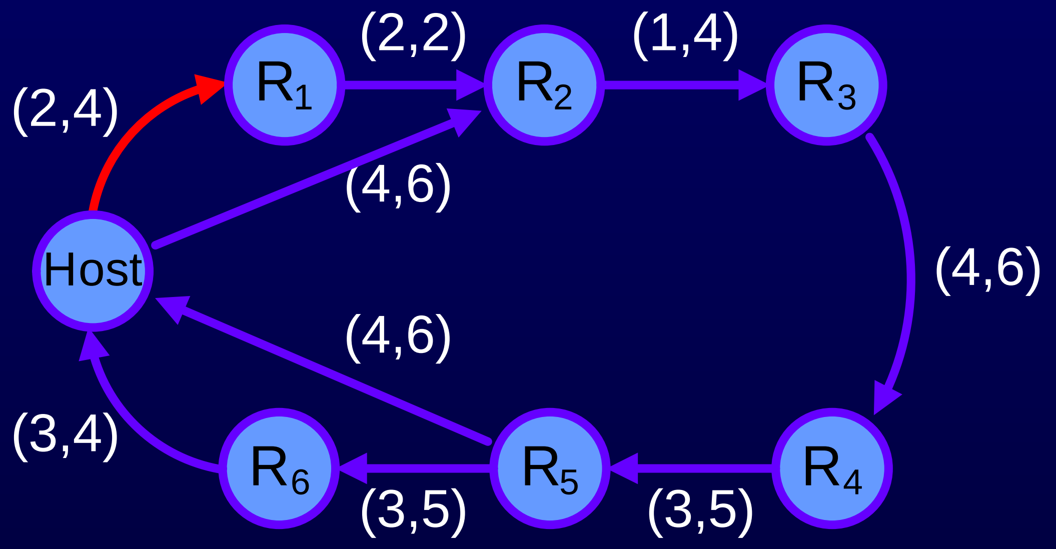


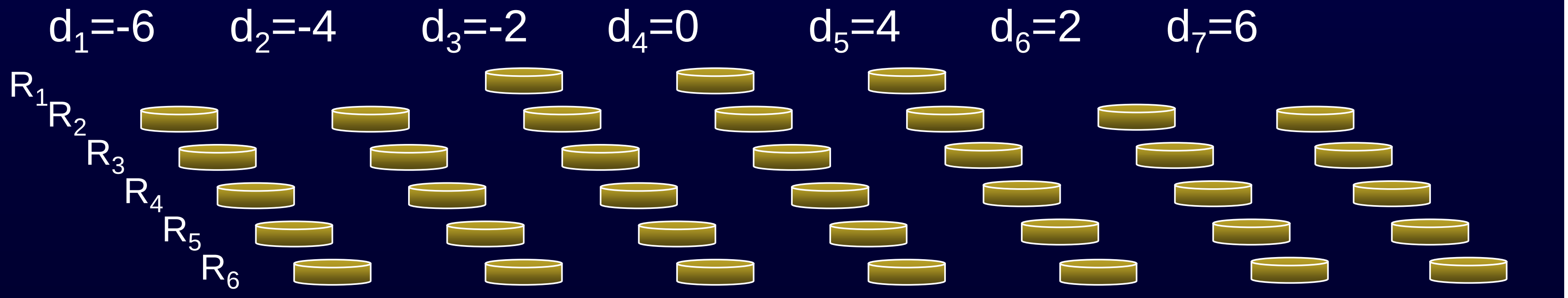
$d_1=-6$ $d_2=-4$ $d_3=-2$ $d_4=0$ $d_5=4$ $d_6=2$ $d_7=6$



An Example

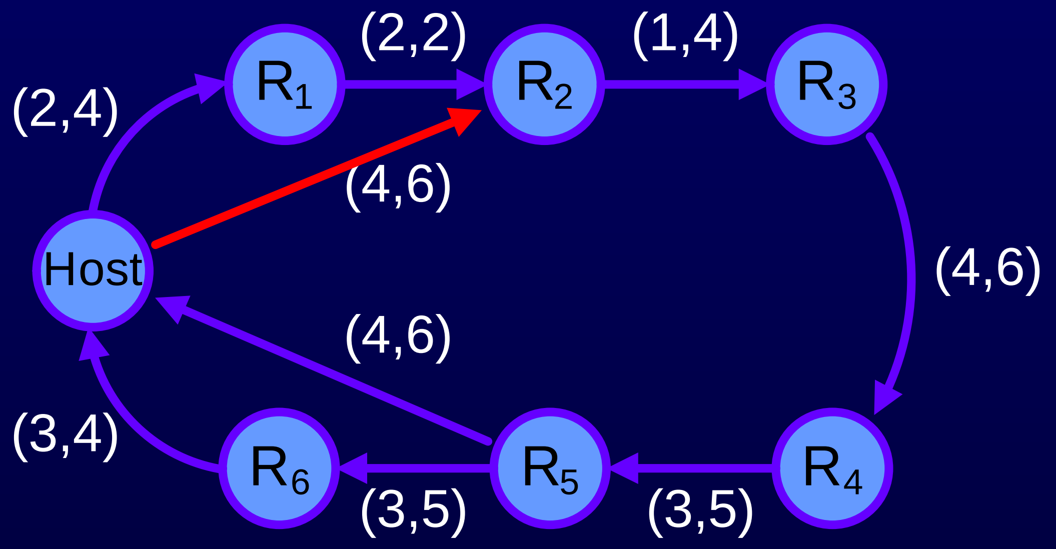
Clock Period = 6

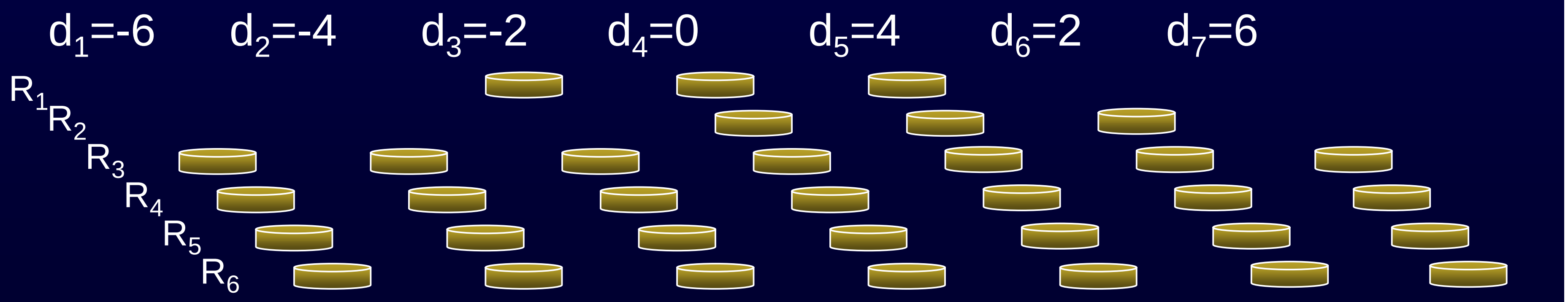




An Example

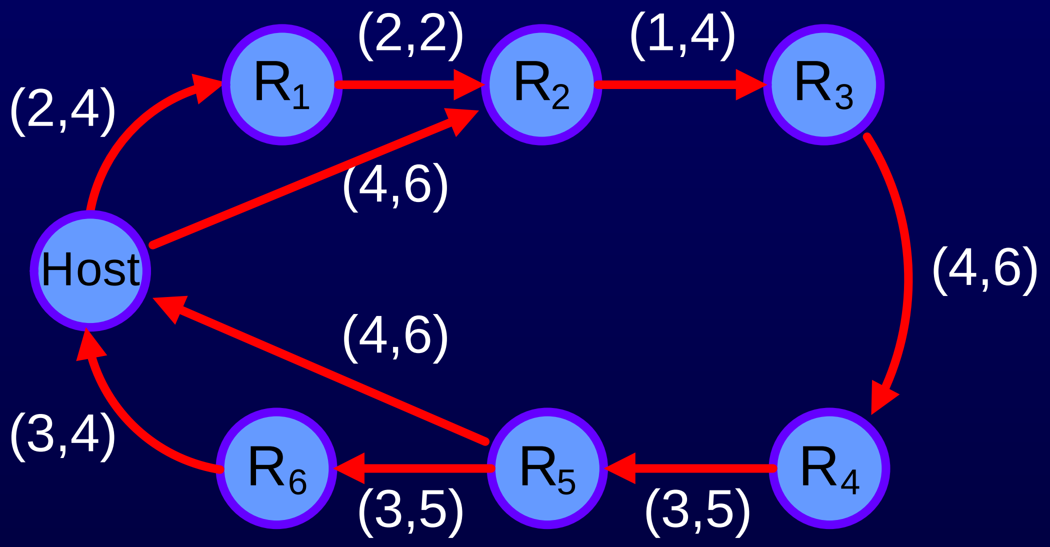
Clock Period = 6



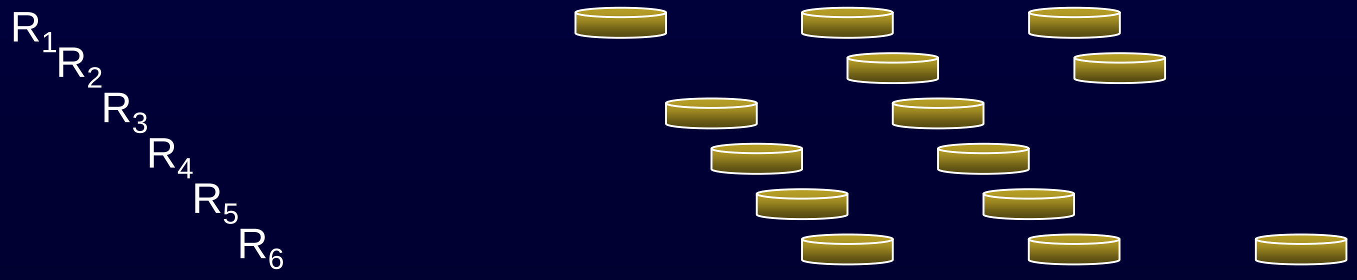


An Example

Clock Period = 6



d_1=-6 d_2=-4 d_3=-2 d_4=0 d_5=4 d_6=2 d_7=6



Outline

- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

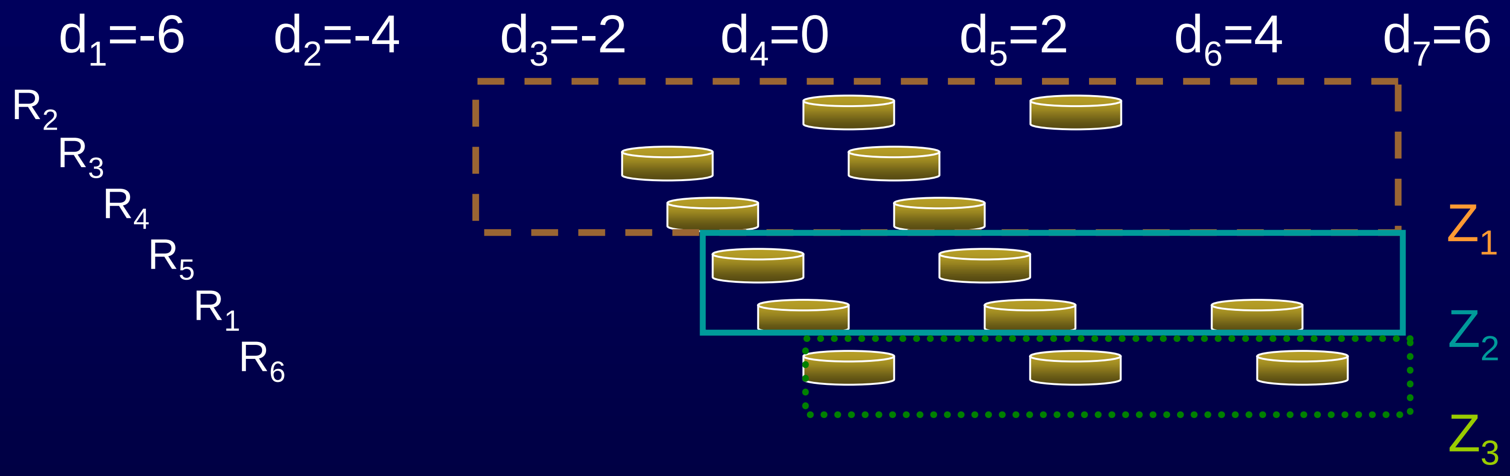
Zone-Based Scheduling

- For a large circuit, the number of binary variables is still very large even though all the redundant binary variables are pruned. We propose a two-phase zone-based approach to solve a large circuit heuristically.
- A zone is defined as a portion of registers in the circuit graph. The maximum number of binary variables associated with a zone is limited. Therefore, each zone can be scheduled efficiently.

First Phase

- Given a target clock period, we use [4] to obtain an initial feasible schedule.
- Then, we partition the registers in the circuit graph into several zones. Note that a register associated with few binary variables has higher priority to be re-scheduled.

First Phase – Zone Partitioning



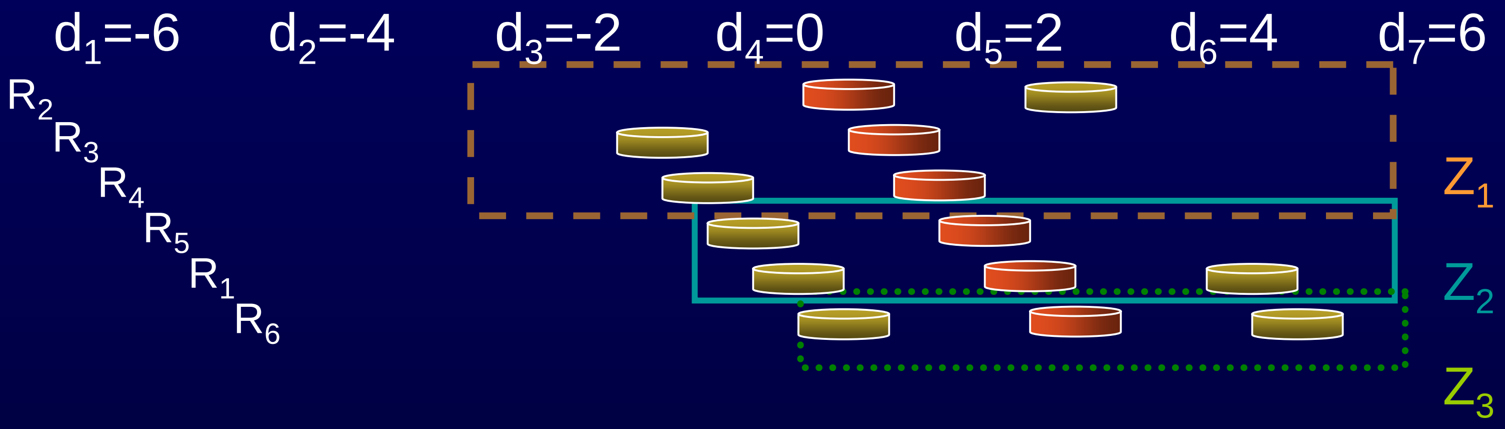
The maximum number of binary variables within a zone : 6

Z_1={R_2,R_3,R_4}

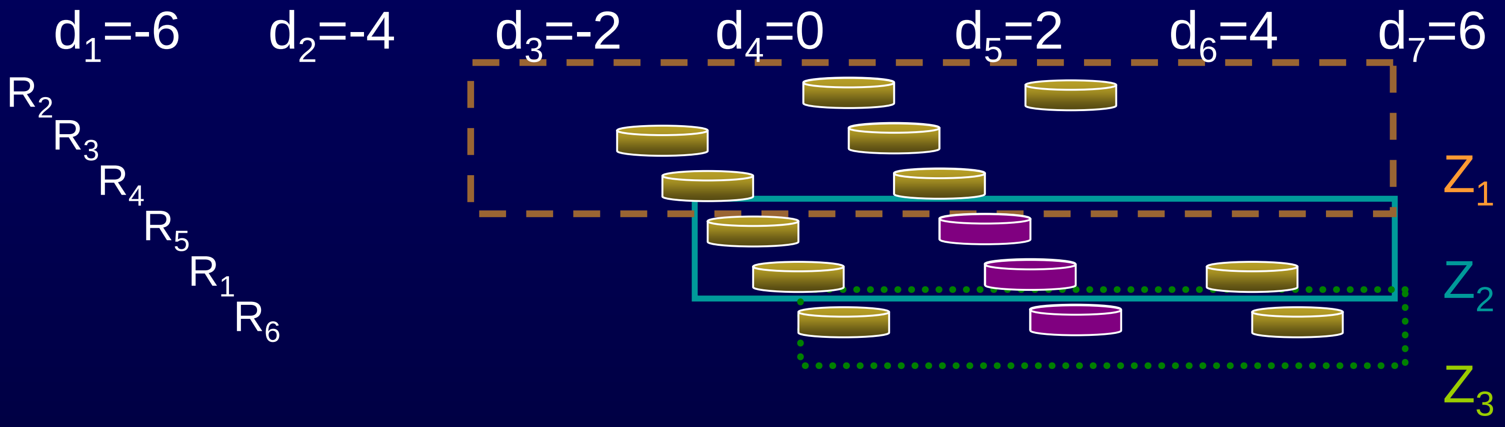
Z_2={R_1,R_5}

Z_3={R_6}

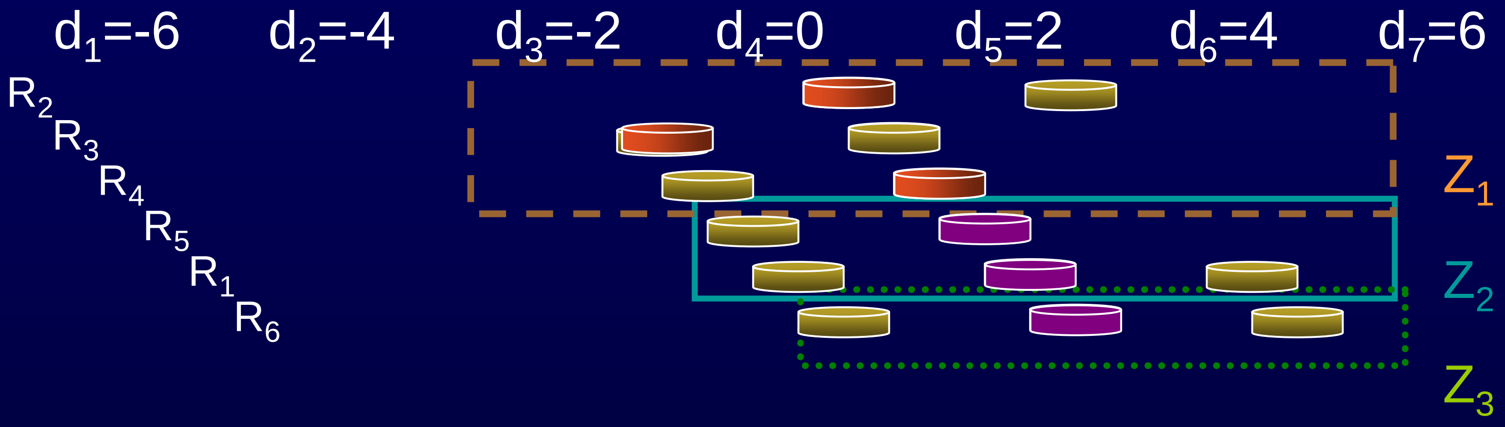
First Phase – Zone 1



First Phase – Zone 1

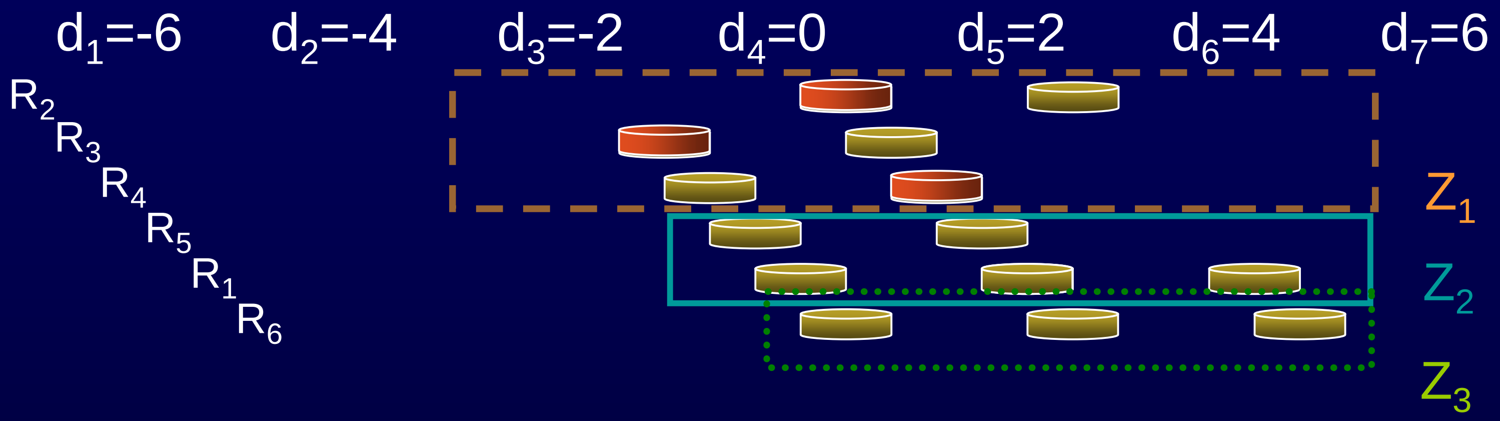


First Phase – Zone 1



Peak_Current = 5

First Phase – Zone 2



d_1=-6 d_2=-4 d_3=-2 d_4=0 d_5=2 d_6=4 d_7=6

R_2

R_3

R_4

R_5

R_1

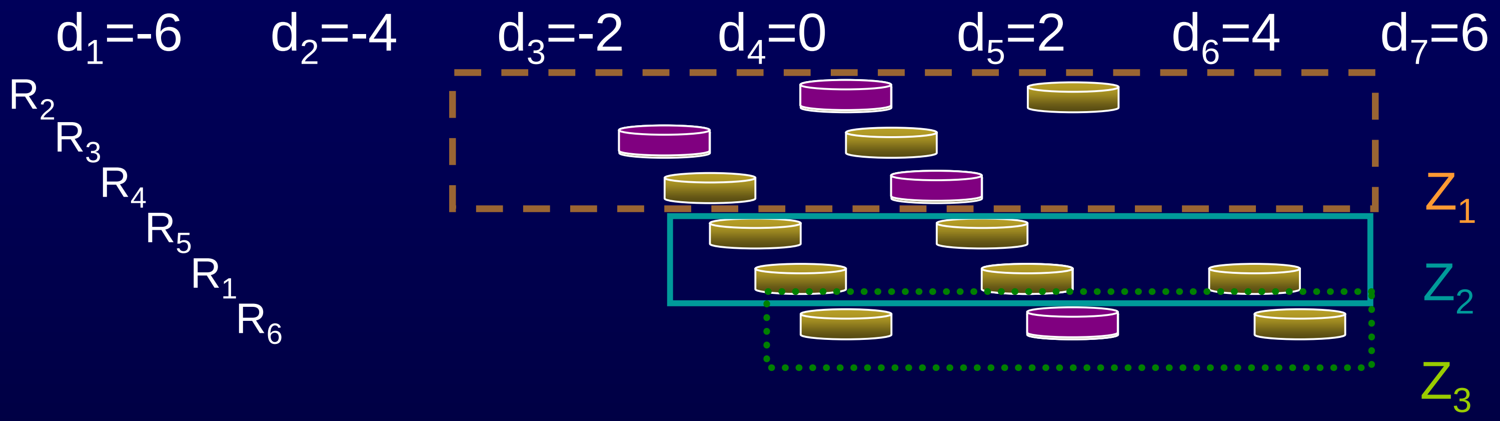
R_6

Z_1

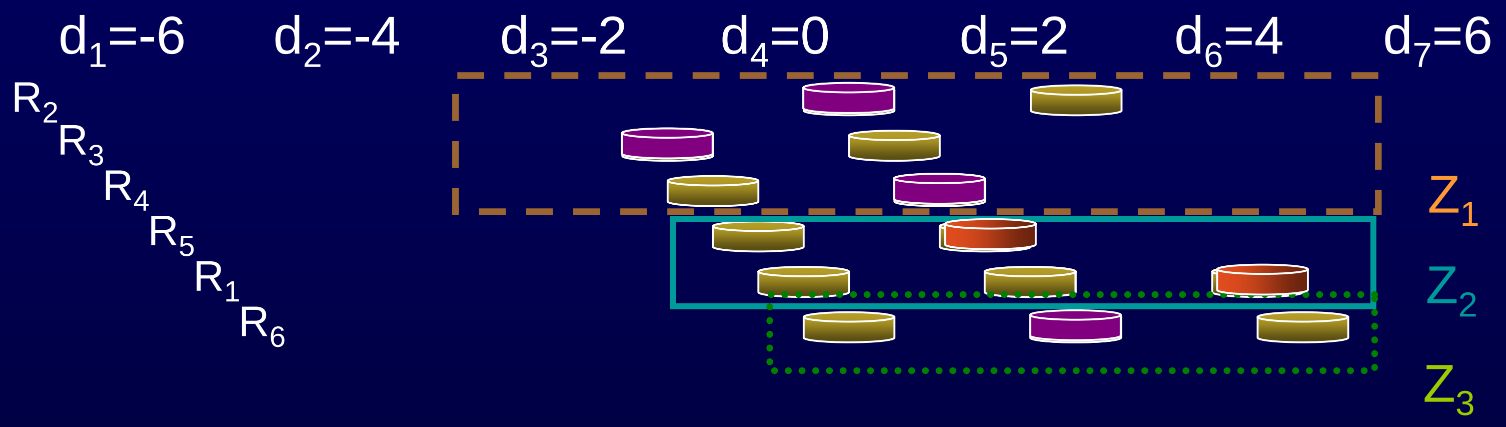
Z_2

Z_3

First Phase – Zone 2

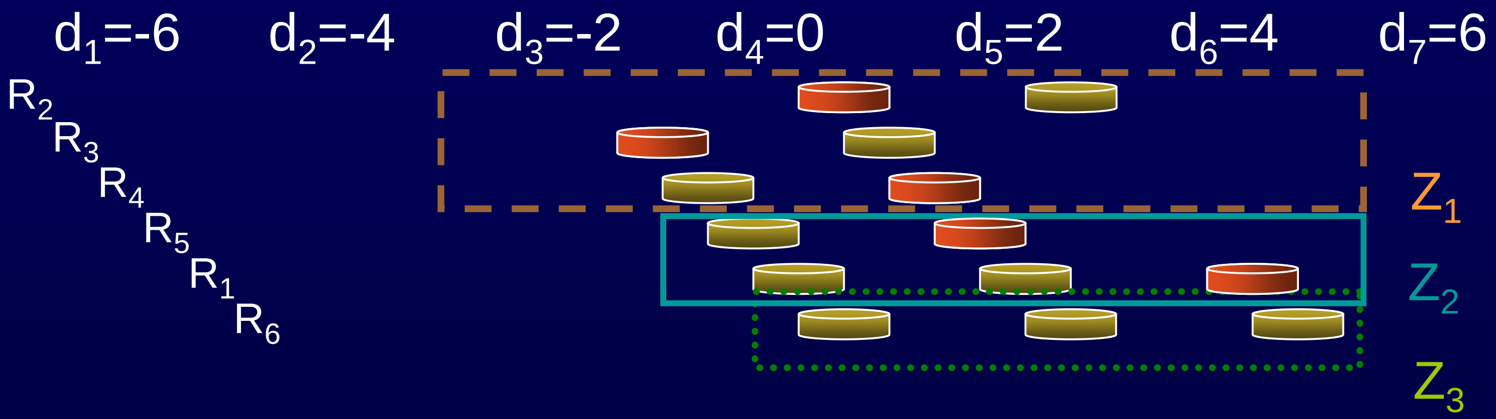


First Phase – Zone 2

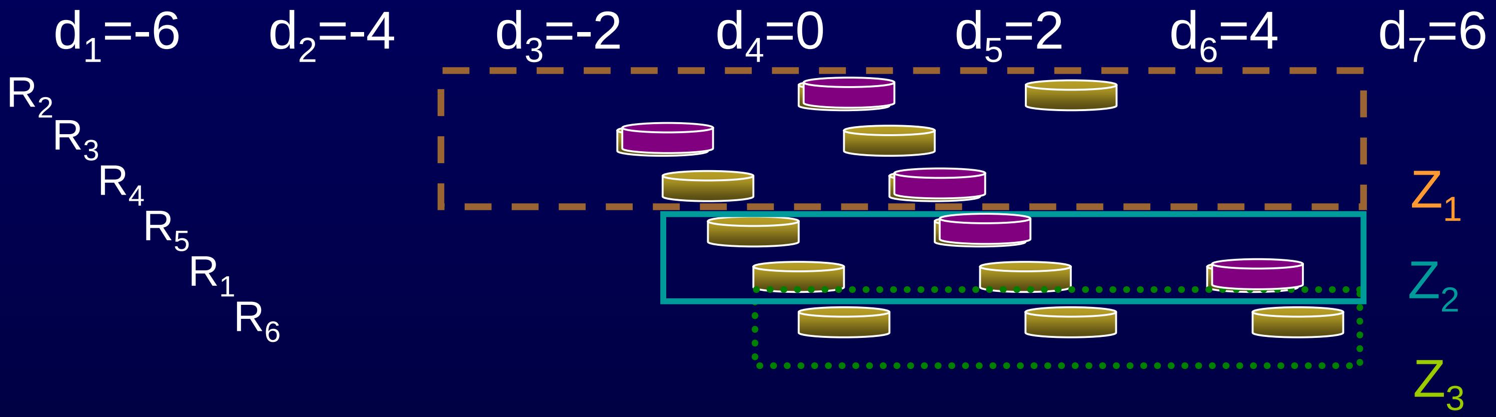


Peak_Current = 4

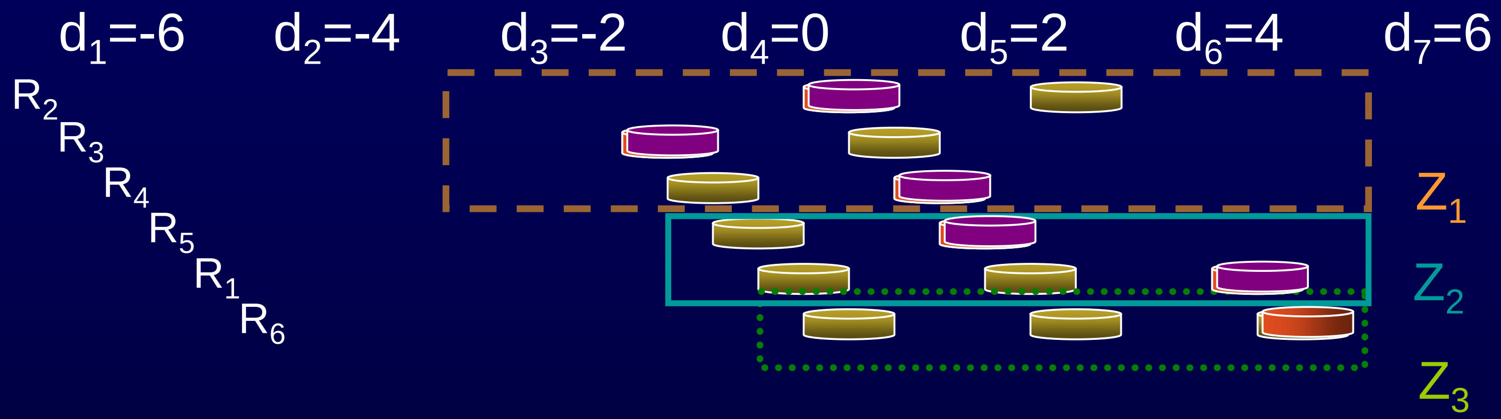
First Phase – Zone 3



First Phase – Zone 3



First Phase – Zone 3

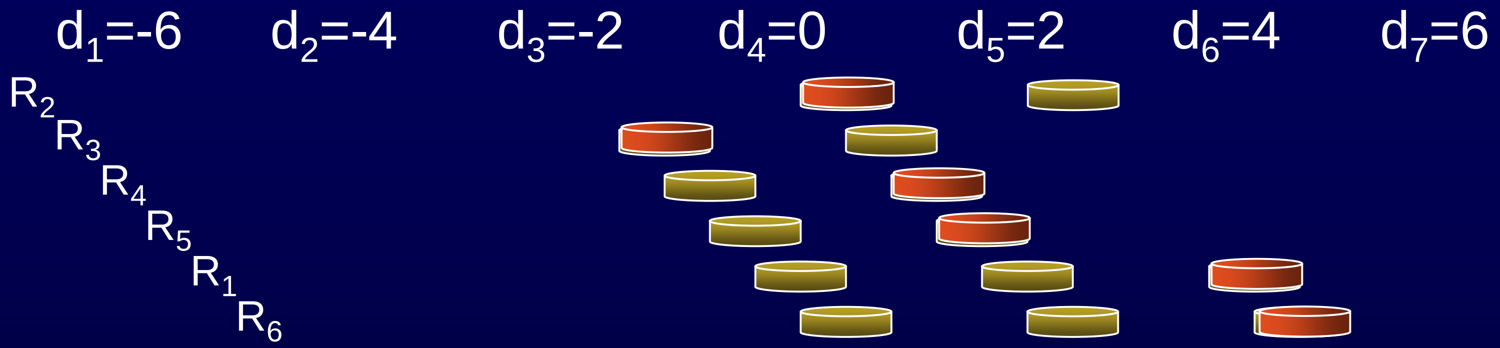


Peak_Current = 3

Second Phase

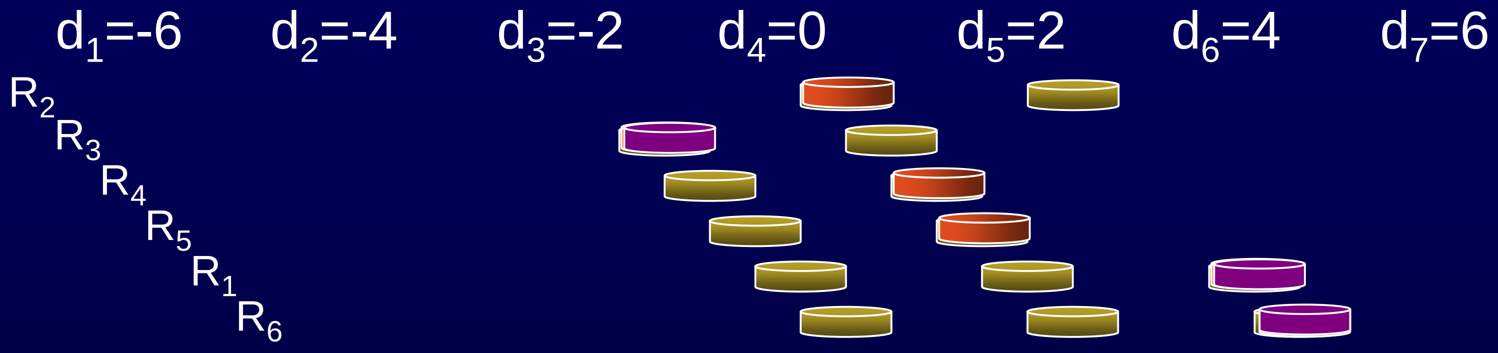
- A clocking domain is called the hottest clocking domain, if and only if the total current derivative at this clocking domain is exactly the same as the peak current.
- The registers scheduled in hottest clocking domains are re-scheduled. The iteration process repeats until the peak current cannot be further reduced.

Second-Phase -- Zone 4



$Z_4=\{R_2,R_4,R_5\}$

Second-Phase -- Zone 4



d_1=-6 d_2=-4 d_3=-2 d_4=0 d_5=2 d_6=4 d_7=6

R_2

R_3

R_4

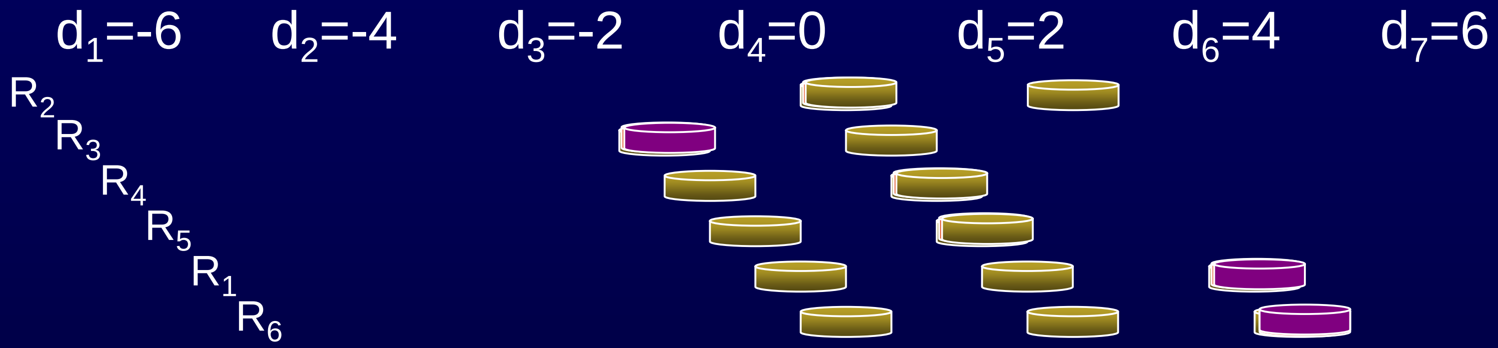
R_5

R_1

R_6

$Z_4=\{R_2,R_4,R_5\}$

Second-Phase -- Zone 4



$d_1=-6$ $d_2=-4$ $d_3=-2$ $d_4=0$ $d_5=2$ $d_6=4$ $d_7=6$

R_2

R_3

R_4

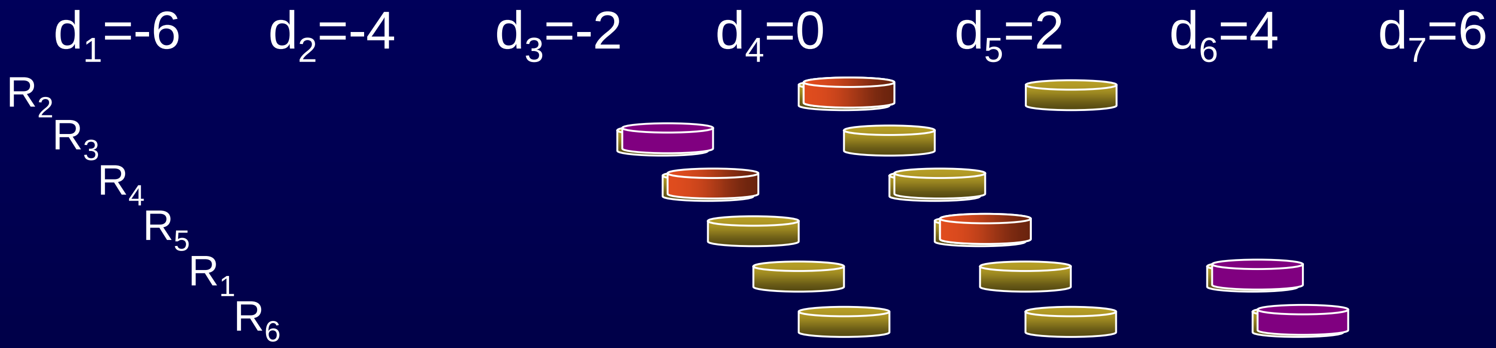
R_5

R_1

R_6

$Z_4=\{R_2,R_4,R_5\}$

Second-Phase -- Zone 4



Z_4={R_2,R_4,R_5}

Peak_Current = 2

Outline

- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

Experimental Results

- The circuits from ISCAS’89 benchmarks suites are targeted to a 0.35 um cell library for the experiments.
- We use Extended LINGO Release 8.0 as the 0-1 ILP solver.
- Running on a personal computer with AMD K8-3GHz CPU and 1G Bytes RAM.

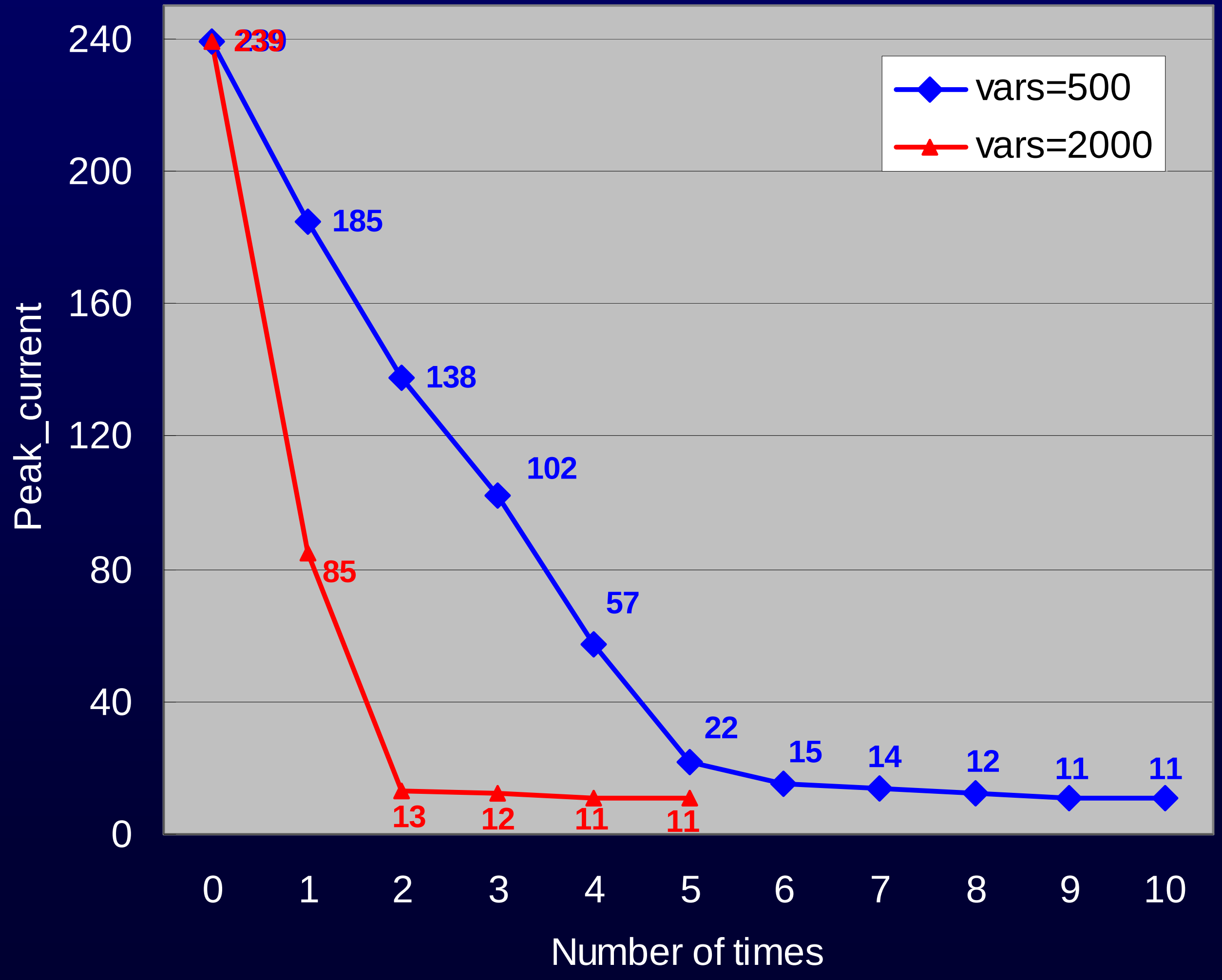
Pruning Redundancies

Circuit	[6]			Reduced Formulations		
	#variables	#constraints	CPU time (s)	#variables	#constraints	CPU time (s)
S3271	6612	2447	30087	3342	2027	88
S3384	24339	4316	--	4917	3531	24
S5378	6265	4860	16046	2088	342	59
S6669	45171	5476	--	3951	3265	122
S9234	14364	6803	--	6554	6042	232
S13207	48837	10062	--	20057	8416	362
S15850	56715	34418	--	18124	28314	--
S35932	60480	12081	714	17516	3714	271
S38584	114708	32189	--	40018	29569	--

Results of Zone Scheduling

Circuit	500 variables in a zone			2000 variables in a zone			whole-circuit	
	peak current	tackled Zones	CPU Time (s)	peak current	tackled Zones	CPU time (s)	peak current	CPU time (s)
S3271	4	8	10	4	4	88	4	88
S3384	5	10	5	5	4	14	5	24
S5378	11	7	7	11	2	54	11	59
S6669	11	10	11	11	5	26	11	122
S9234	7	8	23	7	5	37	7	232
S13207	16	28	205	16	11	285	16	362
S15850	15	29	501	15	12	712	--	--
S35932	288	35	133	288	9	240	288	271
S38584	38	54	334	36	21	513	--	--

Benchmark S6669



Outline

- Introduction
- Preliminaries
- ASAP and ALAP
- Zone-Based Scheduling
- Experimental Results
- Conclusions

Conclusions

- This paper investigates a fast multi-domain clock skew scheduling for peak current reduction.
- The main contribution of our work is that it overcomes the high computational expense of previous work.
- Our approach achieves good results within an acceptable run time.

Thank You!