Behavioral Synthesis with Activating Unused Flip-Flops for Reducing Glitch Power in FPGA

Cheng-Tao Hsieh, Jason Cong, Zhiru Zhang, and Shih-Chieh Chang

National Tsing Hua University, Taiwan

University of California, Los Angeles, USA

Outline

- Firewall registers in FPGA
- Functional error problem
- Low power binding and scheduling
- Experimental results and conclusions

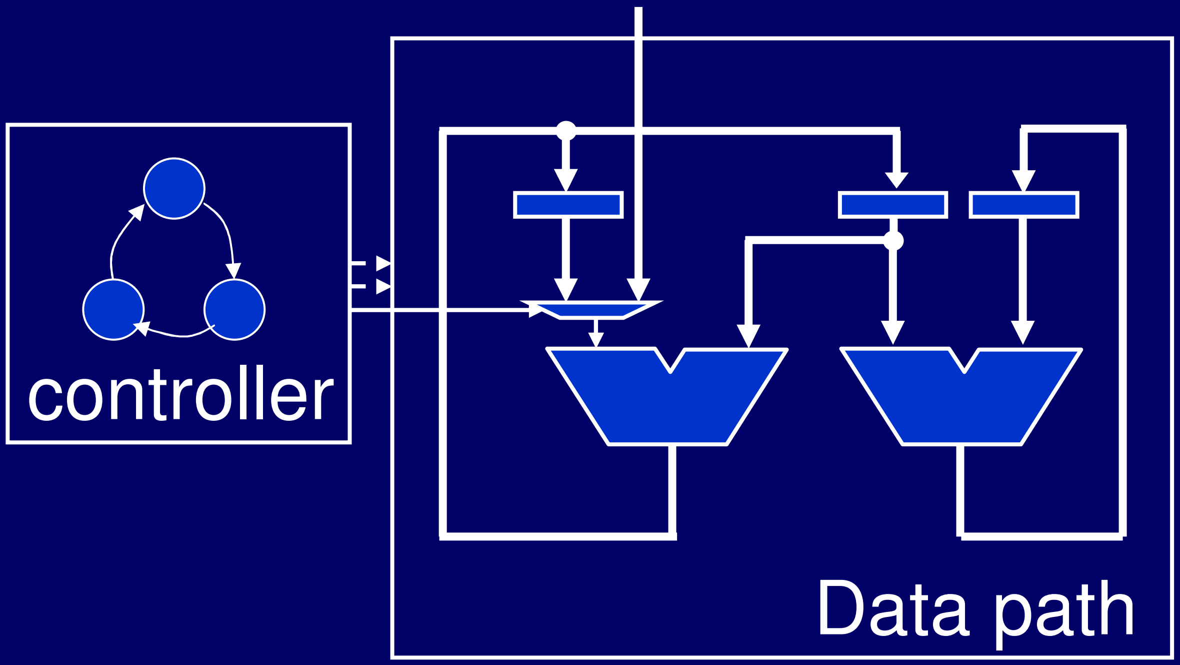
Power in FPGA

- Interconnect dominates FPGA power.
  - [F. Li, etc. in International Symposium on FPGA 2003]
  - [L. Shang, etc. International Symposium on FPGA 2002]

- We modify a design's architecture to reduce certain interconnect's power.

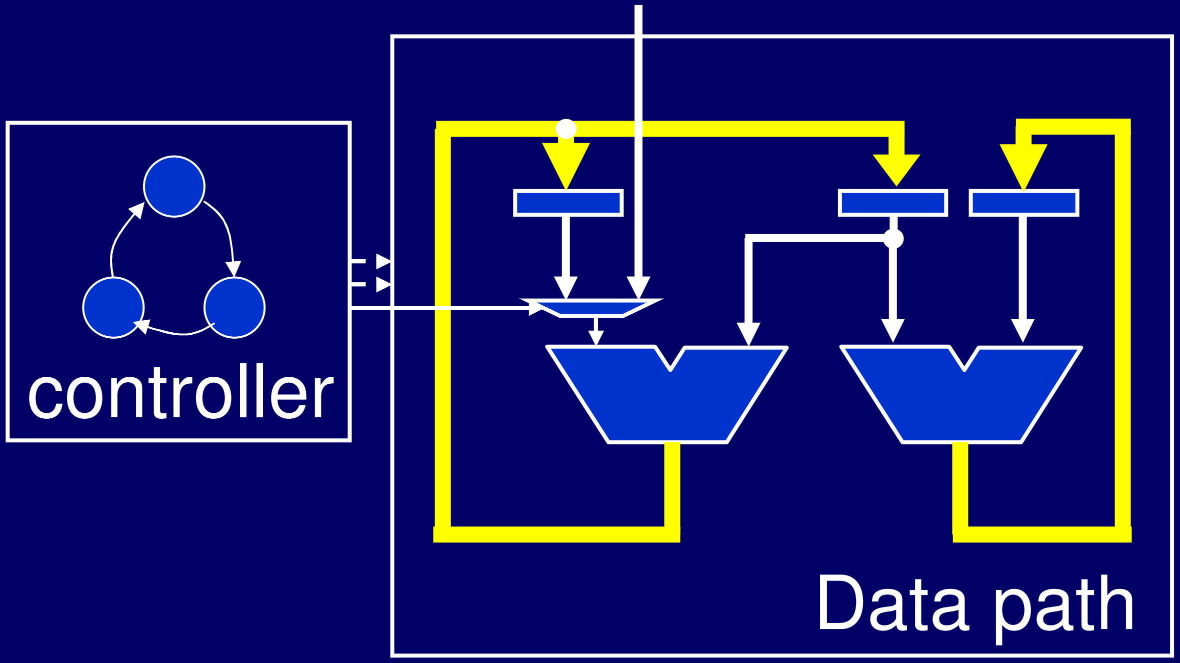
FSMD Architecture

- Finite state machine with data path (FSMD).



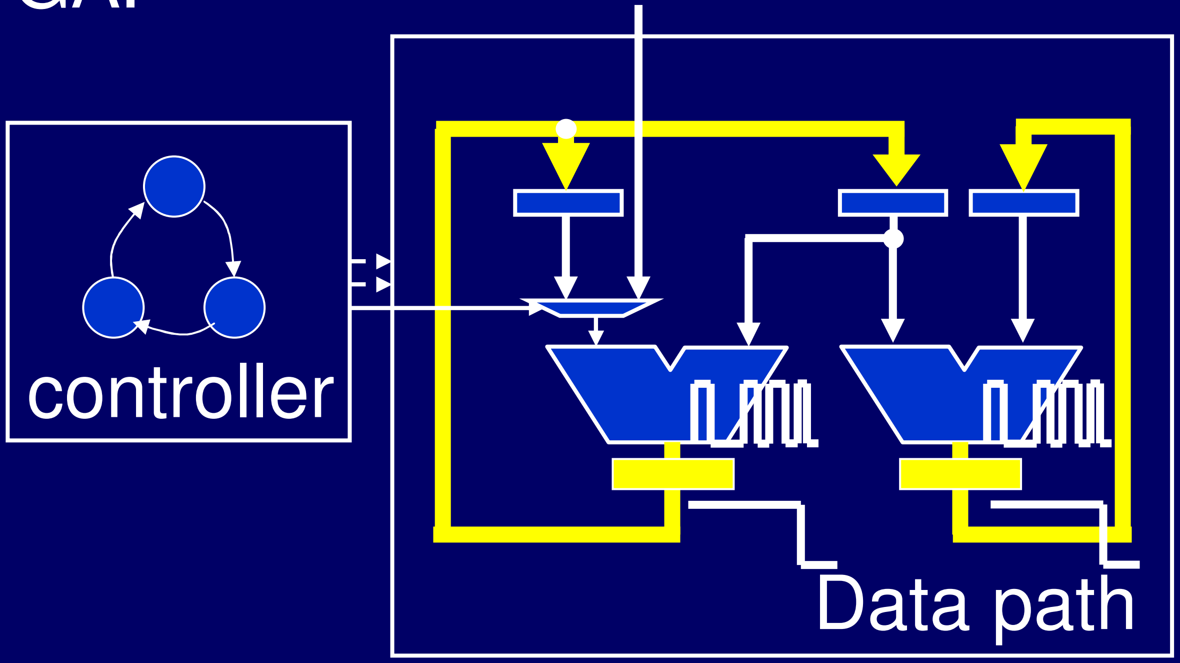
FSMD Architecture

- Finite state machine with data path (FSMD).
- Focus on interconnect power between the output of functional units and the input of registers.



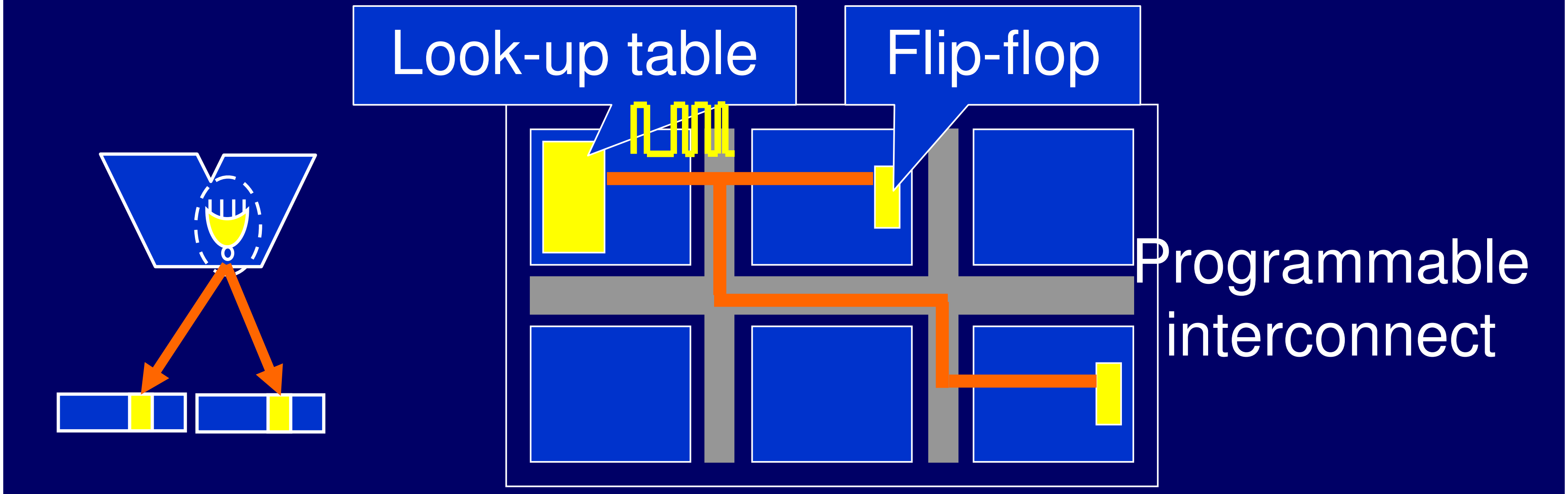
Low Power FSMD

- Insert firewall registers on boundary output signals.
- Glitches occur on a much smaller capacitance in FPGA.



FPGA Implementation

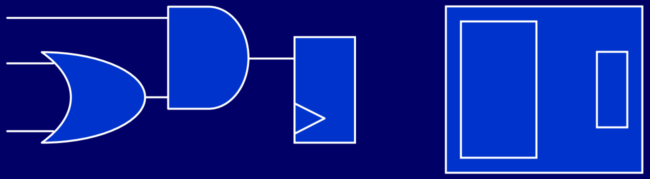
- A *k*-pin net (*k* > 2) is implemented using programmable interconnect which has large capacitance.



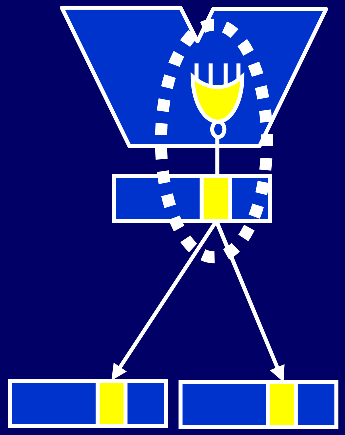
FPGA physical view

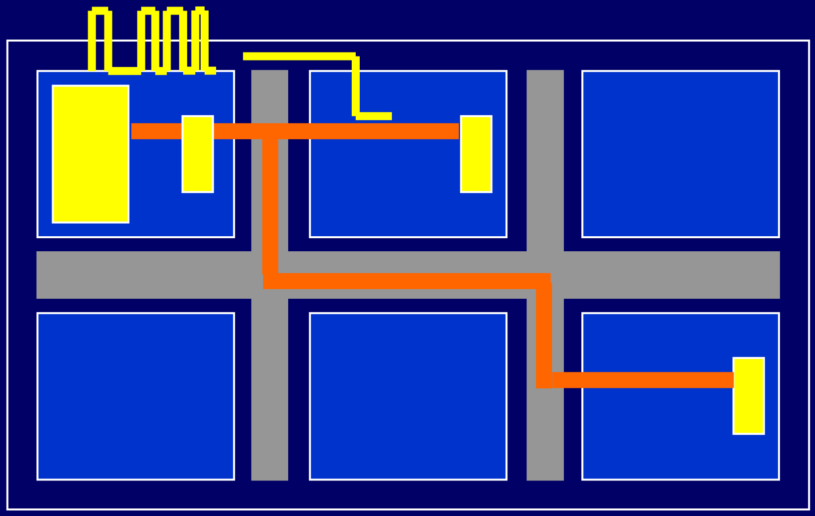
Firewall Register

- Firewall registers can be implemented using registered mode.



Registered mode of configuration

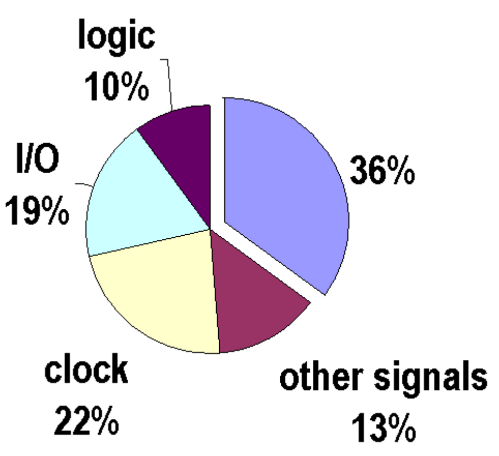


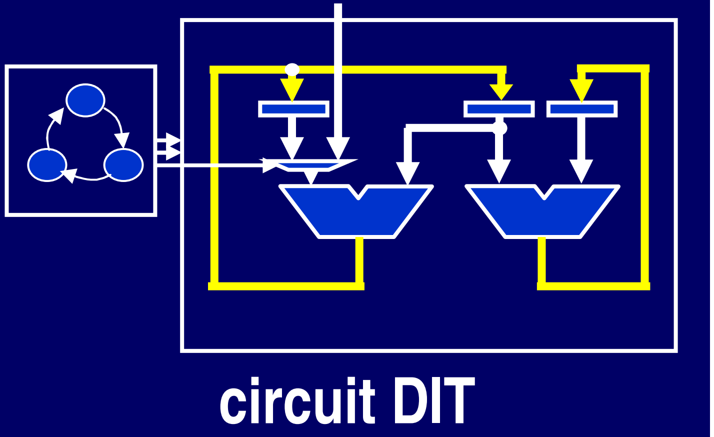


FPGA physical view

Power Profile

- Up to 36% of the total dynamic power in the Xilinx Virtex-II FPGA platform.



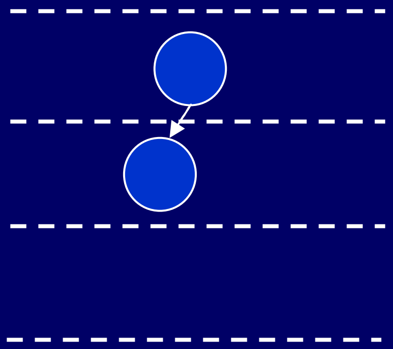


- The reasons:
  - Large glitches
  - Well-optimized common arithmetic units
    - Dedicated 18x18-bit multiplier blocks

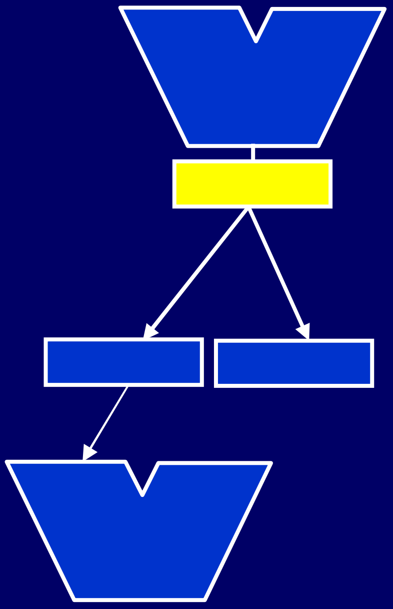
DIF	21%
LEE	16%
WANG	15%
FEIG_DCT	29%

Functional Error Problem

- Firewall register delays data propagation for one cycle.
- We intelligently schedule and bind to avoid hazards.



Partial scheduling



Partial data path

Previous Work

- Activate abundant flip-flops to block glitches → pipeline or retiming.
- Perform optimization on gate-level netlists or physical circuits.
  - Accurate
  - But, not able to solve hazards by scheduling and binding.

Contribution

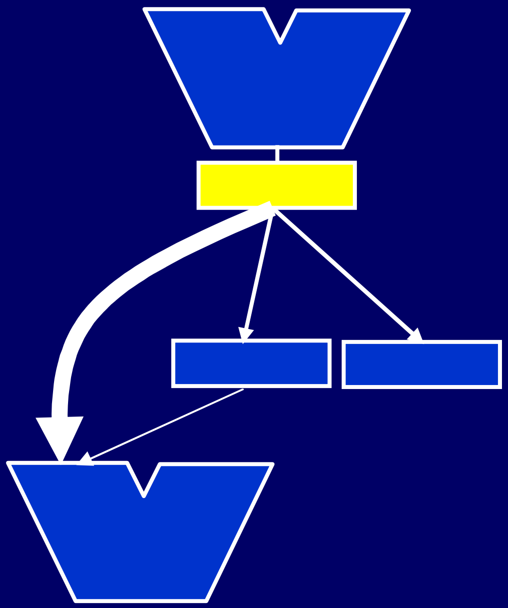
- Extend the solution space by one additional dimension of using/not using firewall register.
- provide methods to guide the insertion of firewall registers in the behavioral synthesis stage
- We have incorporated our techniques into xPilot, introduced by UCLA.

Outline

- Firewall registers in FPGA
- Functional error problem
- Low power binding and scheduling
- Experimental results and conclusions

Forwarding

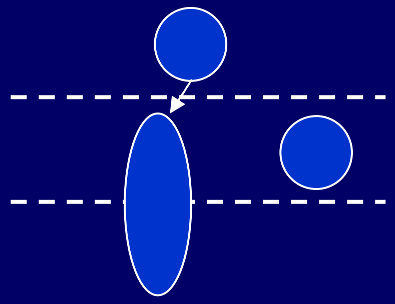
- Use forwarding to solve a partial of functional errors.



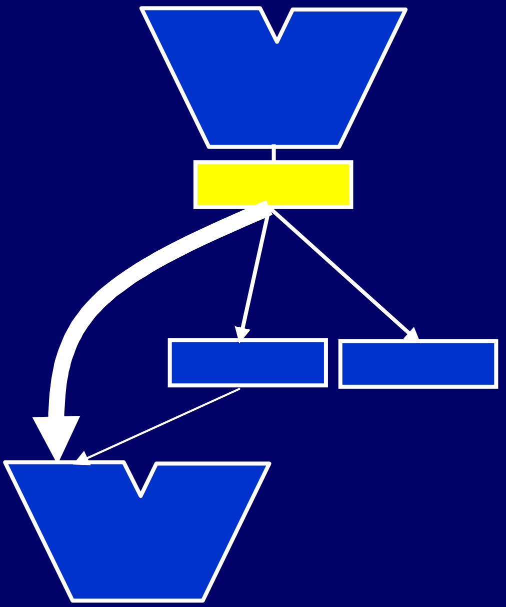
Partial data path

Forwarding to Solve Hazards Completely?

- The consuming is a multi-cycle operation.
- During the reading, an operation writes its result to the firewall register.
  - ➔ Write-After-Read (WAR) hazard

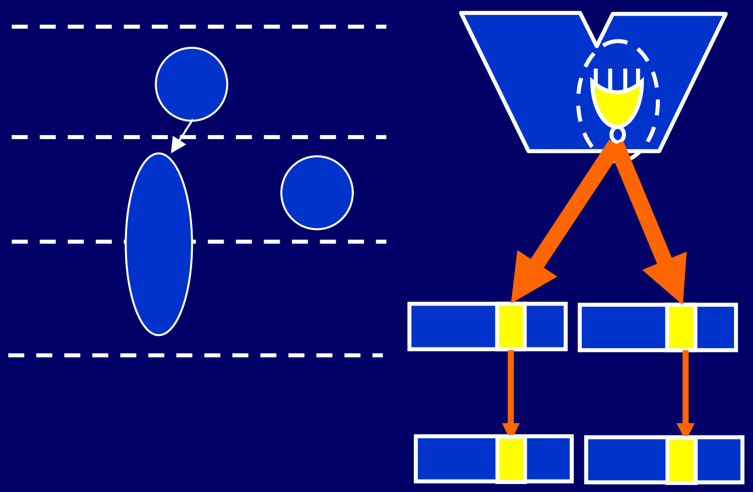


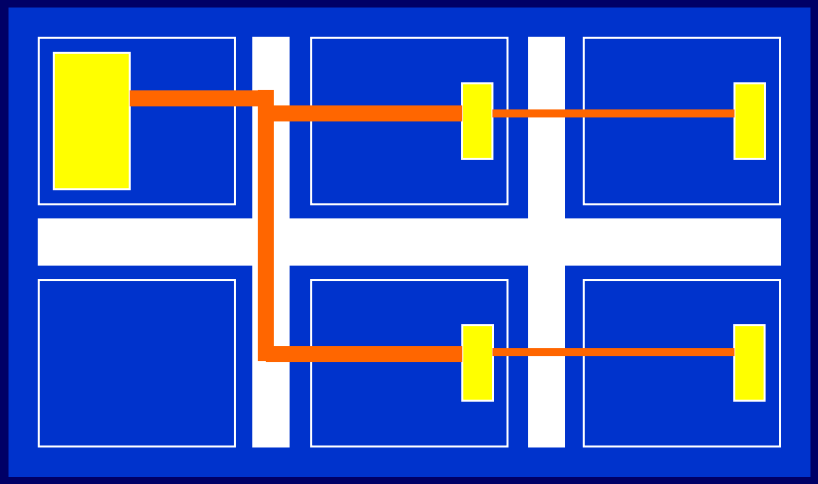
Partial scheduling



Only-one Firewall Register

- Multiple firewall registers cannot use a registered gate.
- No WAR hazards on normal registers.

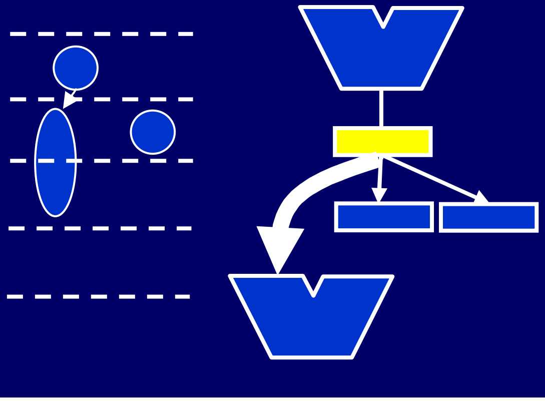




Formal Description

- For a dataflow (*u*, *v*), a WAR hazard occurs if and only if
  - *u* and *v* are scheduled at consecutive cycles
  - ∃ *w* such that *w* and *u* are bound to the same functional unit, and *w* produces results between cycles *i* and *i*+*k*-2.

- Implies that our method is easily applied.



Outline

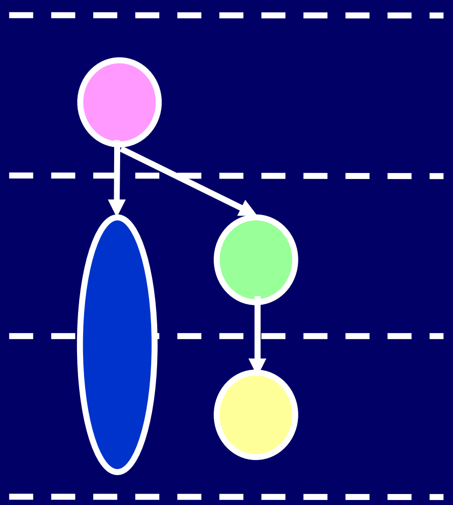
- Firewall registers in FPGA
- Functional error problem
- Low power binding and scheduling
- Experimental results and conclusions

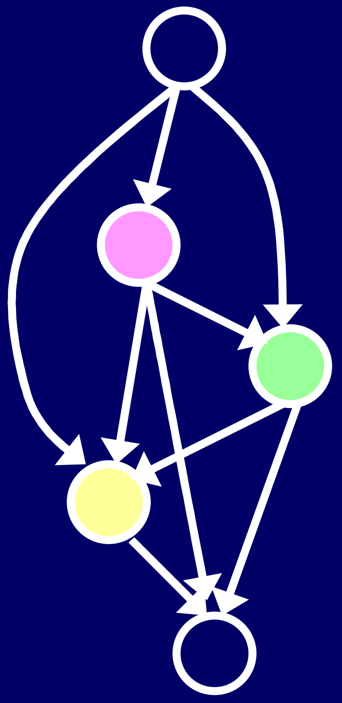
Difficulty of Binding with Firewall Registers

- Goal is to minimize the power.
- Consider switching and firewall register simultaneously.

Network Flow Problem

- Reduce to min-cost N-flow problem.
- Previous work: network flow problem to minimize switching activity.

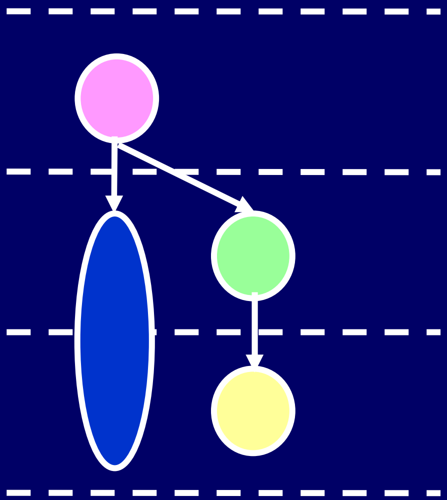


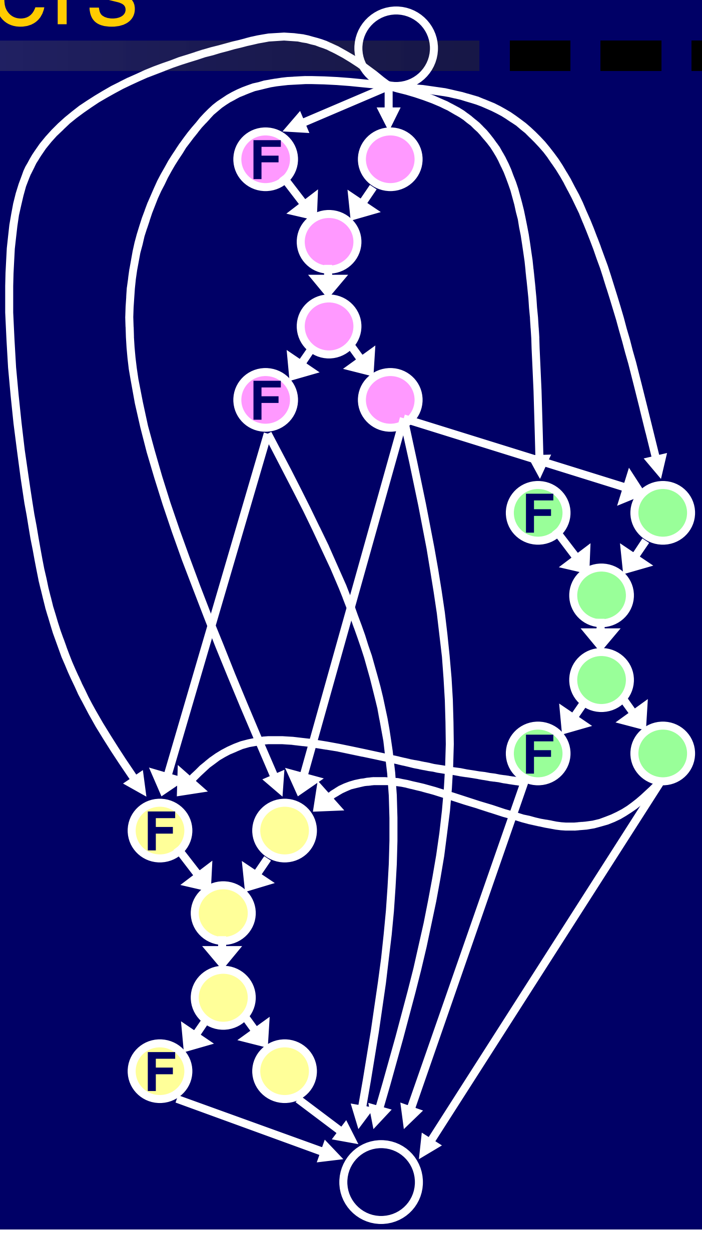


Modifying Network

1. Show the two conditions, FR and non-FR, of a functional unit.
2. Exclusive of FR and non-FR conditions.
3. Guarantee a flow must stay in either FR or non-FR.

Network for Switching and Firewall registers



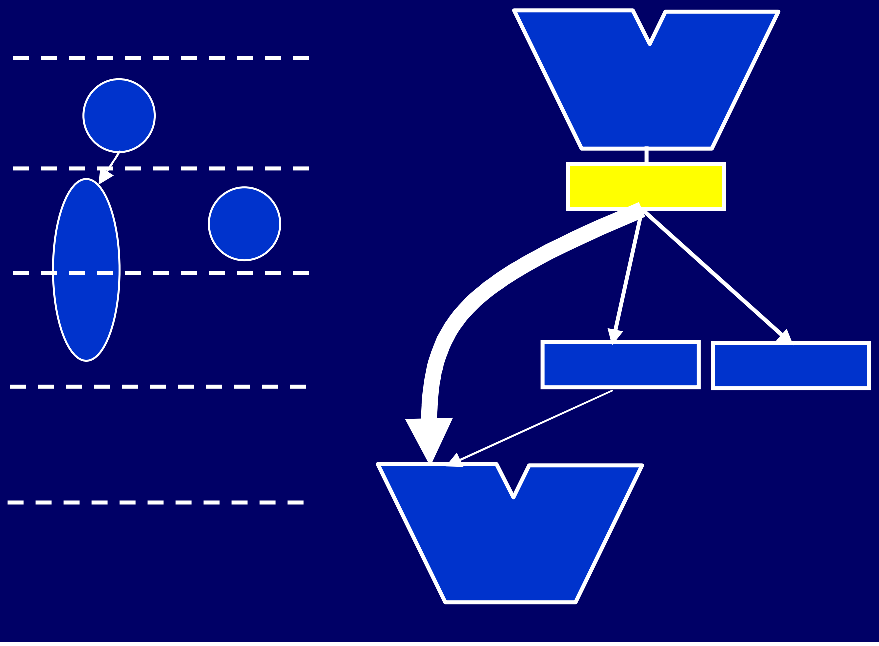


Low-Power Scheduling

- In scheduling, the goal is not to minimize the power.
- We maximize the insertion of firewall registers.

Slack

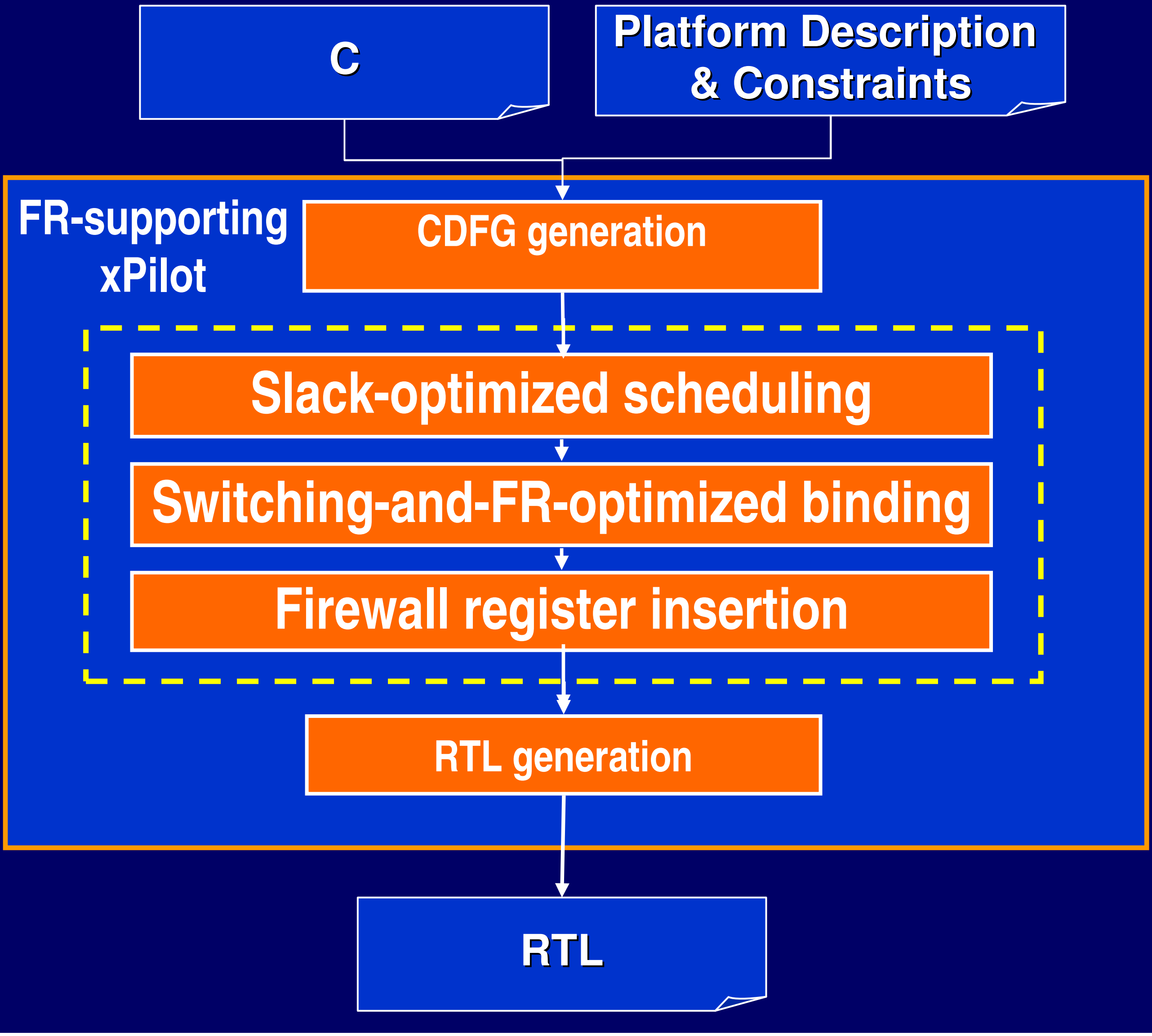
- Slack
  - Zero slack: a hazard potentially occurs.
  - Positive slack: a hazard never occurs.
- Because of latency constraint, this problem is traditionally called the timing budgeting problem.



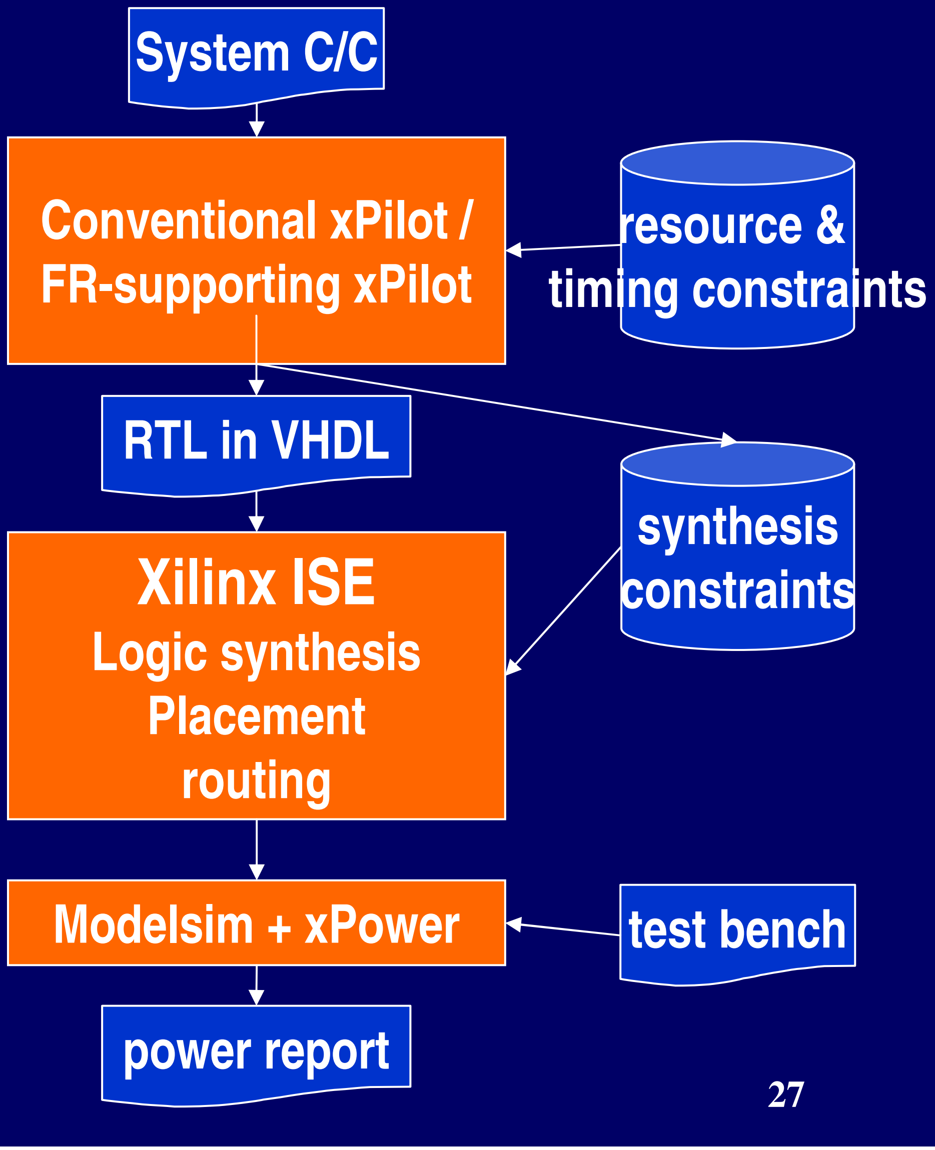
Outline

- Firewall registers in FPGA
- Functional error problem
- Low power binding and scheduling
- Experimental results and conclusions

Incorporating Firewall Register Method into xPilot


C
Platform Description & Constraints
FR-supporting xPilot
CDFG generation
Slack-optimized scheduling
Switching-and-FR-optimized binding
Firewall register insertion
RTL generation
RTL

Experimental Flow



- Device
  - XC2V500 in Xilinx’s Virtex-II family
  - XC2V1500 for benchmark CHEM
- Clock cycle
  - 15ns

Experimental Results

Design	ADD /SUB	MUL	Conventional	Conventional + FR	FR-supporting
			Power (mW)	Power (mW)	Power (mW)
ARAI	6	1	146	128	119
DIF	6	2	174	155	126
DIT	7	3	199	202	140
LEE	6	4	174	158	115
MCM	13	6	241	243	220
WANG	5	4	204	144	133
CHEM	33	33	719	553	556
DIR	11	12	275	195	160
HONDA	9	10	223	175	166
PR	5	3	137	101	86
Avg.			1	0.84	0.72

Experimental Results

Design	ADD /SUB	MUL	Conventional		Conventional + FR		FR-supporting	
			FF	Slice	FF	Slice	FF	Slice
ARAI	6	1	676	833	772	828	788	919
DIF	6	2	788	987	852	988	916	1029
DIT	7	3	932	1214	1012	1173	1076	1296
LEE	6	4	1028	1126	1092	1132	1172	1156
MCM	13	6	1652	2085	1956	2261	1940	2225
WANG	5	4	836	998	916	1025	980	1065
CHEM	33	33	5076	6063	5892	6126	5716	5330
DIR	11	12	1732	2091	2084	2156	2100	2152
HONDA	9	10	1364	1768	1668	1741	1668	1764
PR	5	3	548	859	612	872	676	850
Avg.			1	1	1.13	1.01	1.18	1.04

Conclusions

- Firewall registers
- To avoid WAR hazards, propose an FR-supporting behavioral synthesis flow.
- The experimental results show that the reduction in dynamic power is around 28%.