Scheduling with Integer Time Budgeting for Low-Power Optimization

Wei Jiang, Zhiru Zhang,
Miodrag Potkonjak and Jason Cong

Computer Science Department

University of California, Los Angeles

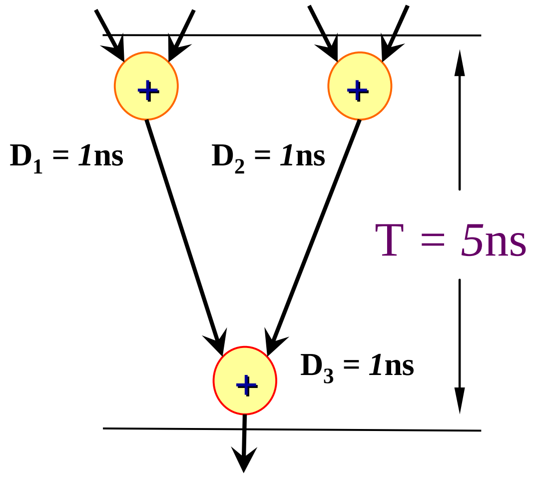
Supported by NSF, SRC.

Outline

- Introduction to integer time budgeting (ITB) problem
- Low-power scheduling
- Experimental results
- Conclusions and future work

Integer Time Budgeting (ITB) Problem

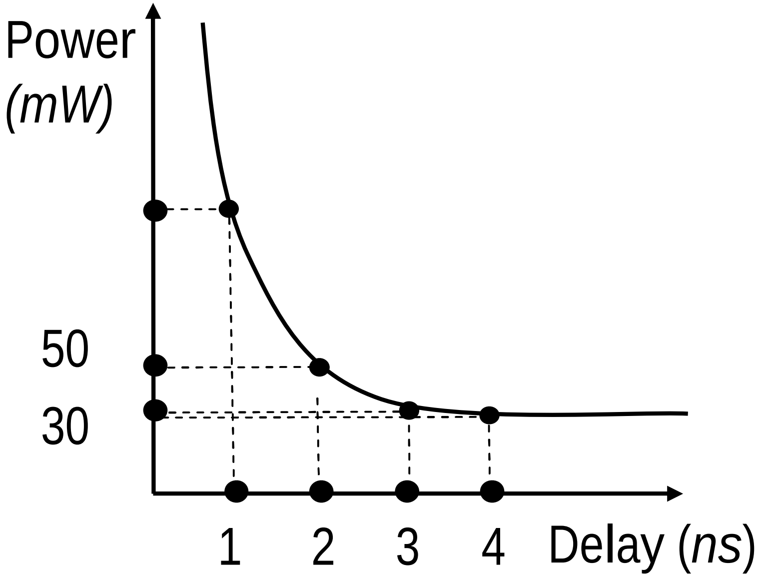
- Definition
  - Slack: the amount of extra delay that each component (either a small gate or a large module) of a design can tolerate without violating the given timing constraint
- Time budgeting problem
  - The problem of distributing the slacks to different modules of a design to optimize some objectives (such as area, power)
  - Example: reduce area/power by using slower adders
- Integer time budgeting problem
  - The slacks must be integral values
  - Applications: scheduling, gate sizing, interconnect planning



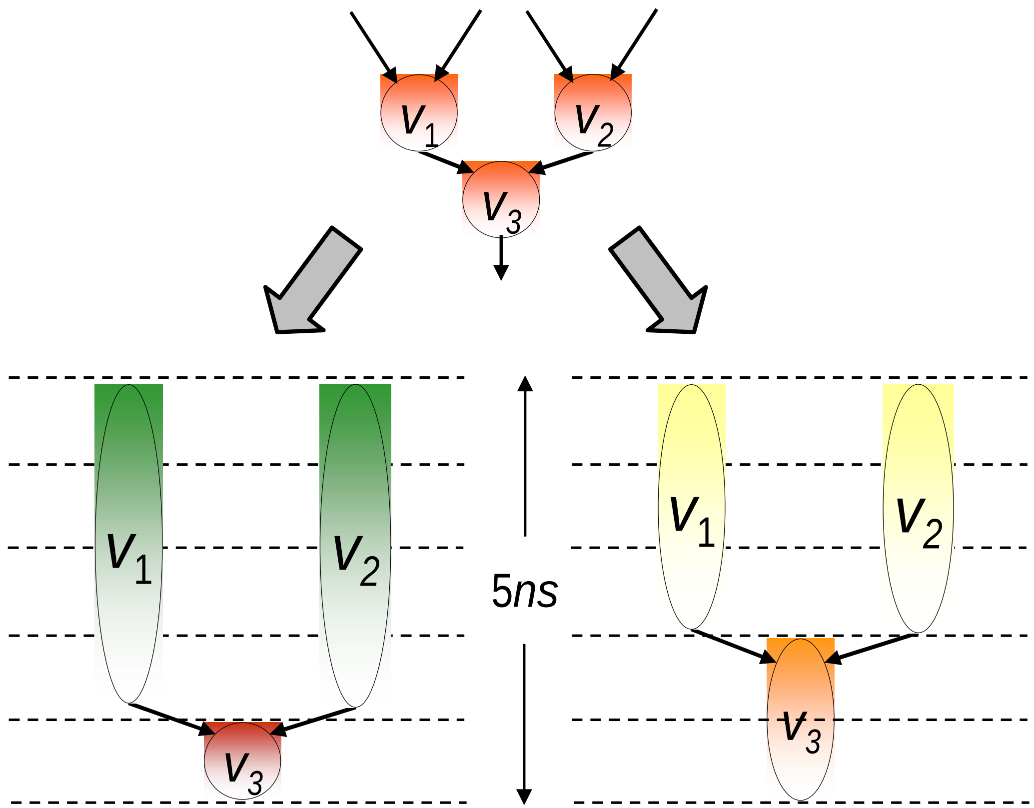
Slacks: S_1=S_2=S_3=3ns

Motivation

- Maximizing sum of weighted slack might be suboptimal for power optimization



Power-delay tradeoff-curve



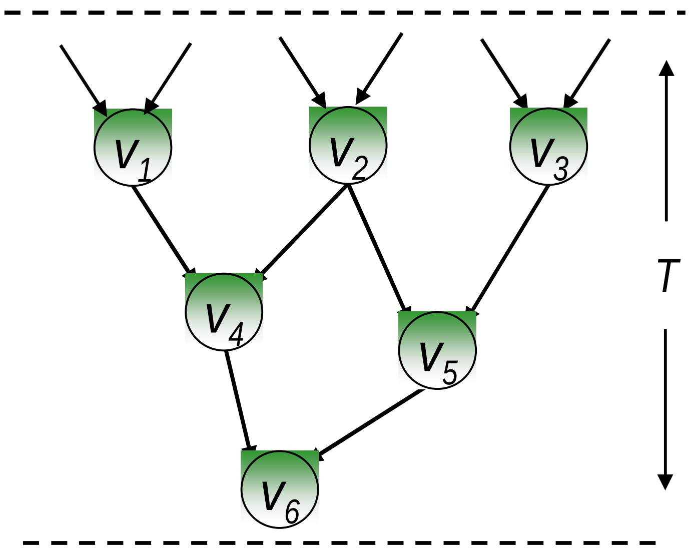
Total slack = 6*ns*
Total power = 160*mW*

Total slack = 5*ns*
Total power = 110*mW*

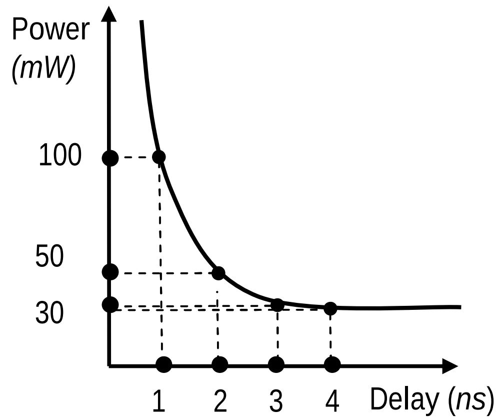
Related Work

- Network-flow-based algorithm
  - A Unified Theory of Time Budget Management, [Ghiasi et al., ICCAD’04]
    - Can solve the ITB problem optimally with linear objective function
  - Design Closure Driven Delay Relaxation Based on Convex Cost Network Flow [Lin et al., DATE’07]
    - Can handle convex objective function
- Mathematical programming approach
  - A Mathematical Formulation of Integer Time Budgeting Problem [Cong et al., TECHCON‘07]
  - Handles convex objective function
  - Intuitive and easy to be incorporated into systems with application-specific design constraints

A Mathematical Formulation of Time Budgeting Problem



Directed Acyclic Graph: $G = (V, E)$



Each f_i is a single-variable convex function

s_i : start time of node v_i

d_i : minimum latency of v_i

b_i : time budget at node v_i

Linearly constrained separable convex optimization problem

$$Min \sum_{i=1}^{|V|} f_i(b_i)$$

$$Subject\ \ to:$$

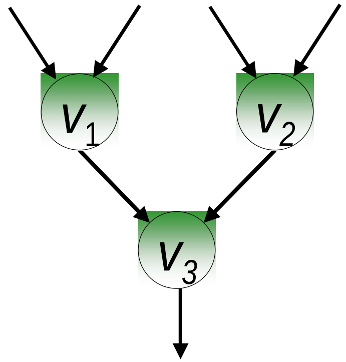
$$s_i + b_i \leq s_j \quad e(i, j) \in E$$

$$b_i \geq d_i \quad \forall v_i \in V$$

$$s_i \geq 0 \quad \forall v_i \in PIs$$

$$s_i \leq T \quad \forall v_i \in POs$$

Totally Unimodular Constraint Matrix



$$
\begin{array}{r c}
 & \begin{array}{cccccc} s_1 & s_2 & s_3 & b_1 & b_2 & b_3 \end{array} \\
\begin{array}{r} s_1 + d_1 - s_3 \le 0 \\ s_2 + d_2 - s_3 \le 0 \\ s_1 \ge 0 \\ s_2 \ge 0 \\ s_3 \le T \end{array} &
\left(\begin{array}{cccccc} 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right) \\
 & \begin{array}{cccccc} J_1 & & J_1 & J_2 & J_1 & \end{array}
\end{array}
$$

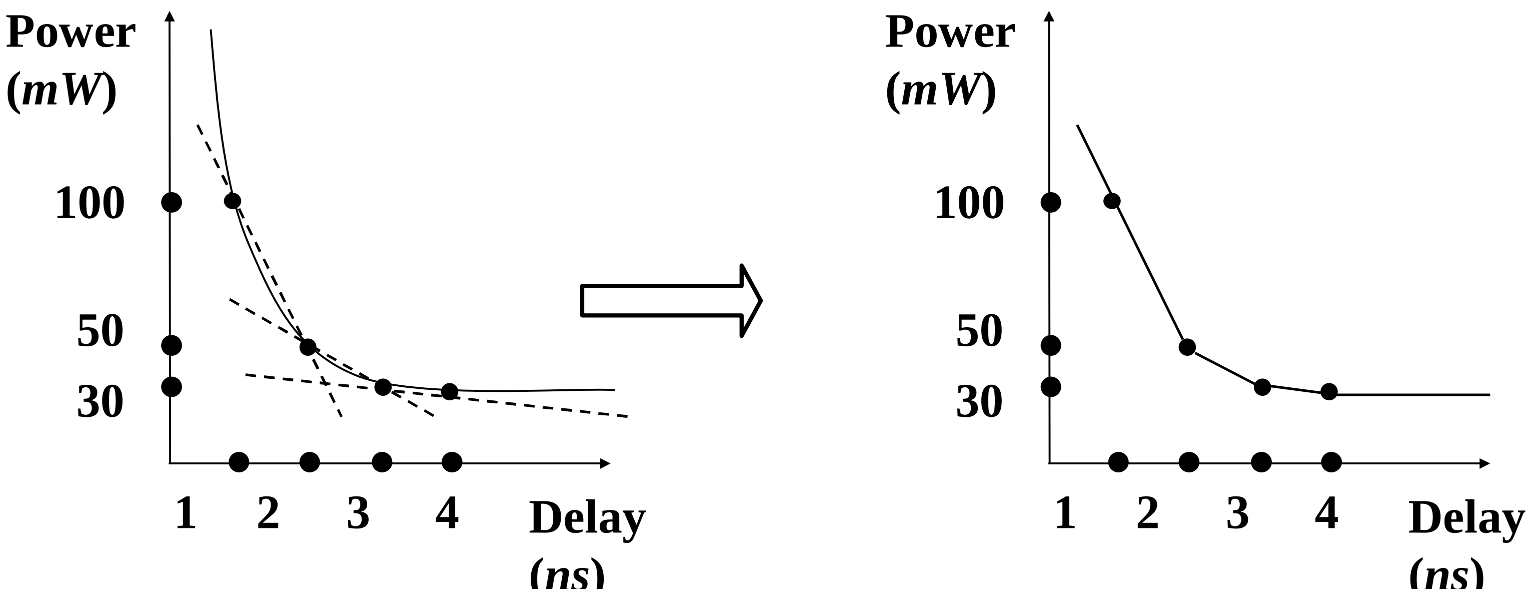
LEMMA 1 (GHOUILA-HOURI [6]). *A* $0, \pm 1$ *matrix* A *is totally unimodular if and only if each subset J of the columns can be partitioned into two classes* J_1 *and* J_2 *such that for each row i, we have* $\left|\sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij}\right| \le 1$.

$$J_1 = \{ j \in J \mid (j > n/2) \wedge (\exists_i a_{ij} = 1) \wedge (j - n/2 \in J) \}$$
$$J_2 = J - J_1$$

$\Rightarrow$ **Theorem 1:**
The ITB constraint matrix is a TUM

Optimizing Separable Convex Objective

LEMMA 2 (HOCHBAUM AND SHANTHIKUMAR [7]). *The linearly constrained, integer separable convex programming problem can be solved optimally in polynomial time with a totally unimodular constraint matrix.*



THEOREM 2 (BASED ON MILLER AND WOLSEY [12]).
If constraint matrix A is totally unimodular, and the objective is a separable piecewise-linear function, we can solve the integer convex separable optimization problem optimally in polynomial time using linear programming relaxation.

Outline

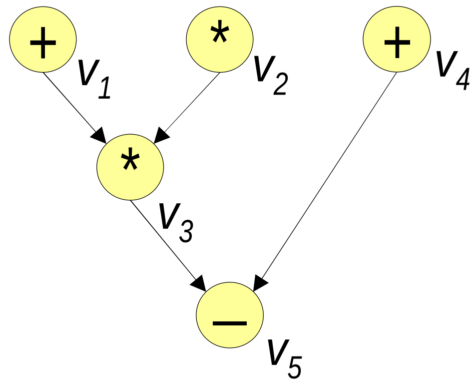
- Introduction to integer time budgeting (ITB) problem
- Low-power scheduling
- Experimental results
- Conclusions and future work

Application to Low-Power Scheduling

- Motivation
  - Scheduling and time budgeting are highly correlated
- Problem
  - Consider the scheduling and budgeting problem together to minimize the average power under time constraint *T*
- Main idea
  - Integrate our ITB problem with the SDC based scheduling [Cong and Zhang, An efficient and Versatile Scheduling Algorithm Based on SDC Formulation, DAC'06]

Low-power Scheduling Problem Formulation

- Given:
  - A data flow graph G
  - A latency constraint T
  - A set of optional scheduling constraints including cycle time constraint, relative timing constraints and resource constraints.
  - A set of power-delay tradeoff curves for each type of operation such as addition, multiplication, etc.
- Objective
  - Get a valid scheduling which satisfies all the constraints and minimize the total power



DFG Example

Low-Power Scheduling

- Each node $v_i \in V_{op}$ is associated with a node budgeting variable $bv(v_i)$ which denotes the # of clock cycles that operation v_i lasts in the final schedule
- Adjust the following constraints
  - Data dependence constraint
    - $\forall (u, v) \in E_d : sv_{beg}(u) + bv(u) \leq sv_{beg}(v)$
  - Latency constraint T
    - $\forall v \in V_{op} : sv_{beg}(u) + bv(v) \leq T$
  - Throughput constraint with initiation interval II
    - $\forall v \in V_{op} : bv(v) \leq II$
- Optimizing total node power $\boxed{Min \sum_{i=1}^{|V_{op}|} pw_{op(v_i)}(bv(v_i))}$
  - We can optimally minimize the total node power in polynomial time

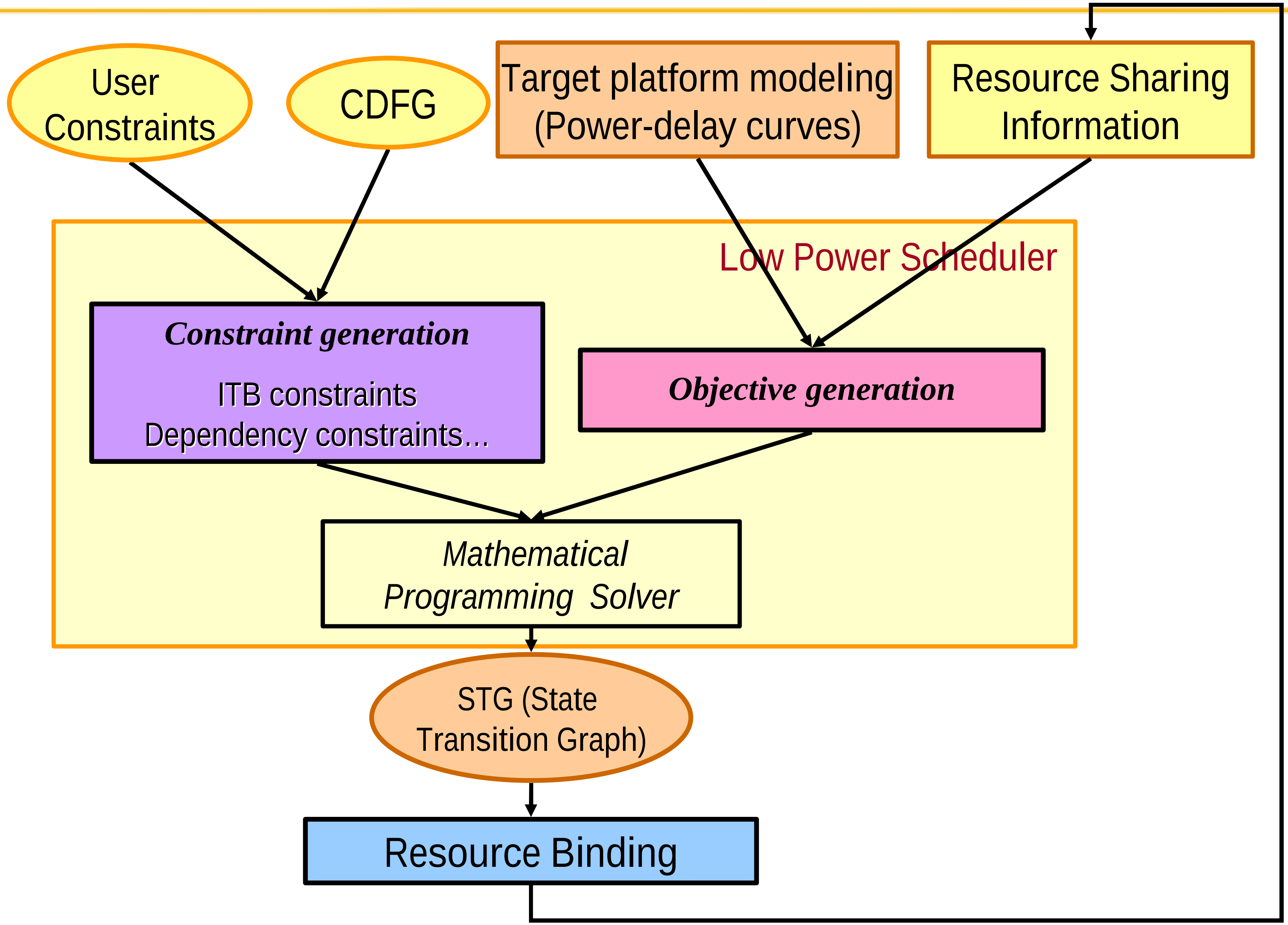
Consideration of Resource Binding

- Optimizing total FU power
  $$Min \sum_{j=1}^{|F|} | f_j | * pw_{op(f_j)}(bv(f_j))$$
  - Constraint matrix is no longer totally unimodular with the requirement that:
    - all operations sharing a same function unit must have same slacks
  - The problem is NP-complete (reduction from 3-SAT)

- Proposed heuristic
  - First solve the continuous version and obtain the "optimal" fractional budget *fb*(v_i) for each node v_i
  - Perform a global rounding by minimizing the least-squares error
    - Objective function is separable convex

$$Min \sum_{i=1}^{|V_{op}|} (bv(v_i) - fb(v_i))^2$$

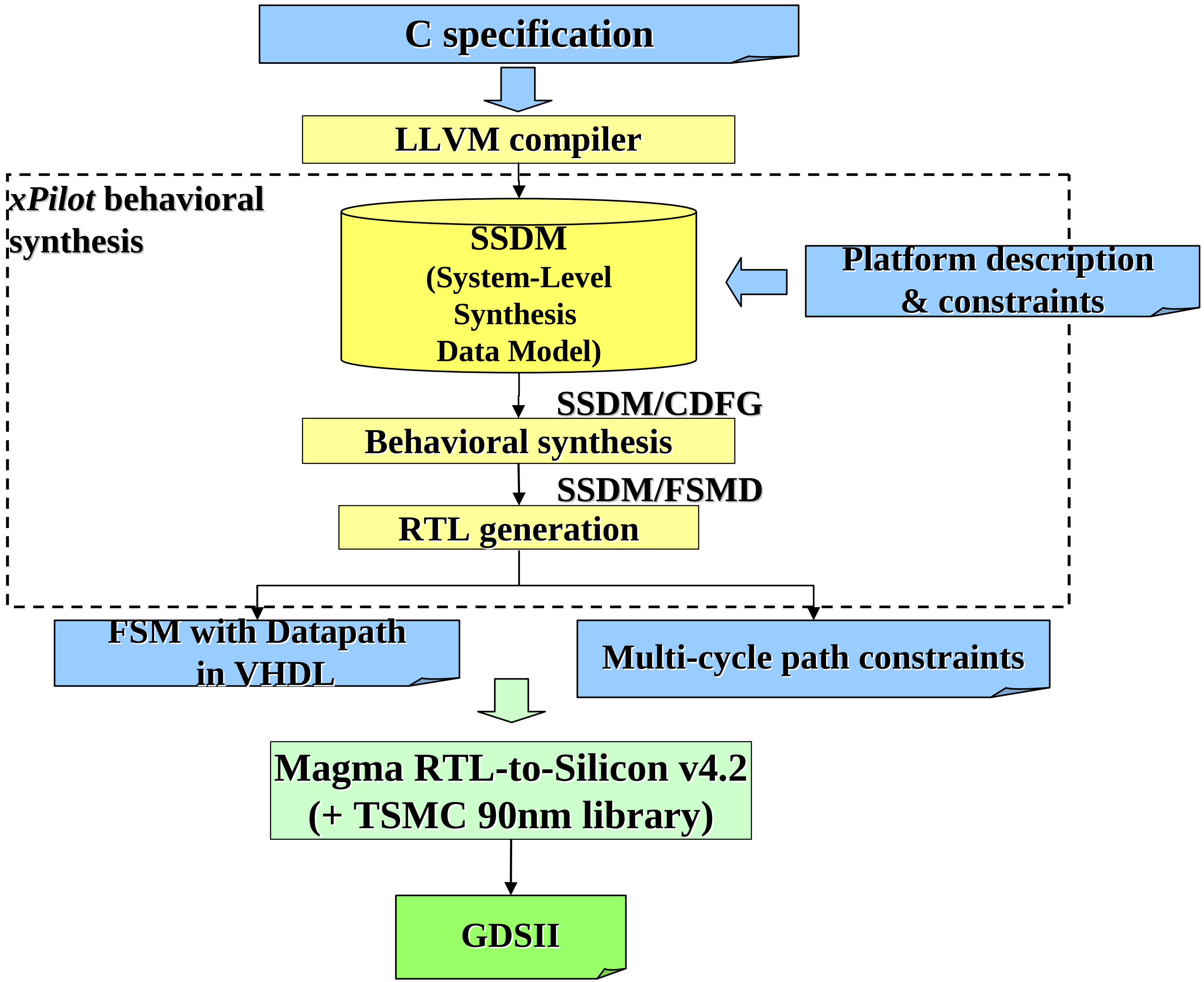
Low-power Scheduling Flow



Outline

- Introduction to integer time budgeting (ITB) problem
- Low-power scheduling
- Experimental results
- Conclusions and future work

C-to-


C specification
LLVM compiler
xPilot behavioral synthesis
SSDM
(System-Level
Synthesis
Data Model)
Platform description
& constraints
SSDM/CDFG
Behavioral synthesis
SSDM/FSMD
RTL generation
FSM with Datapath
in VHDL
Multi-cycle path constraints
Magma RTL-to-Silicon v4.2
(+ TSMC 90nm library)
GDSII

Experimental Results – Comparison with Max Weighted Slack

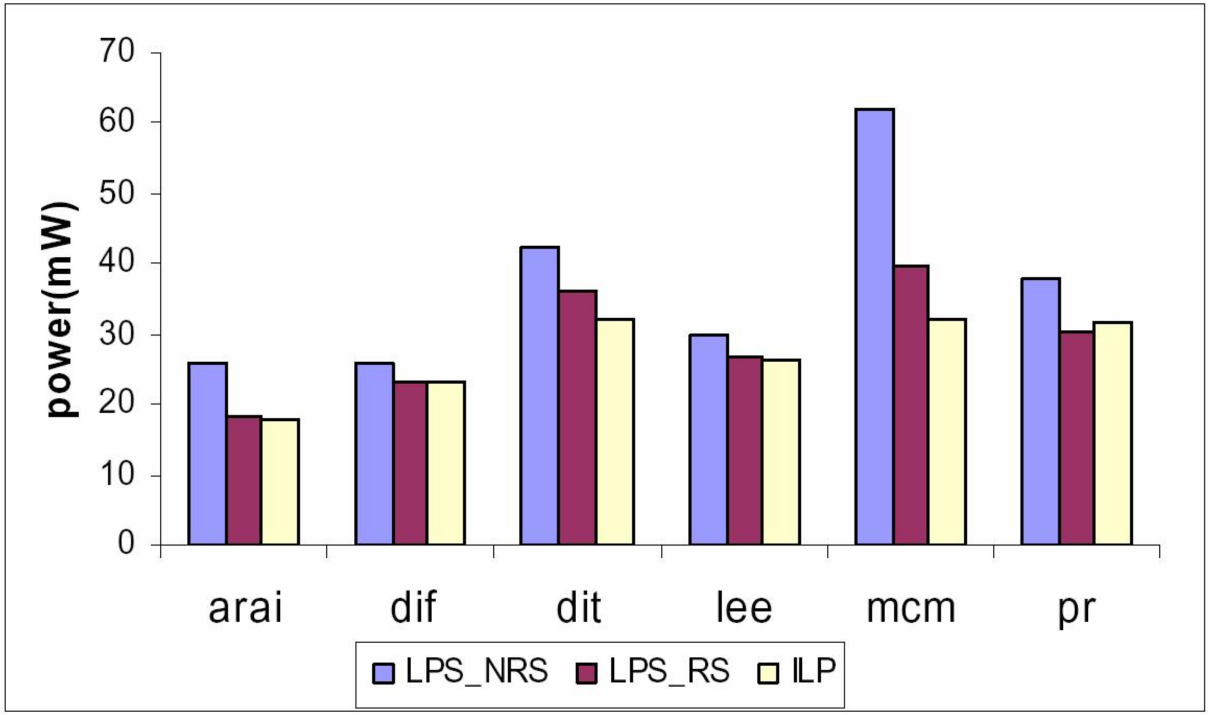
- LPS: Minimize total node power
- WMS: Maximize maximum weighted slack

	Power			Area			Cycle Time		
Design	**WMS** (mW)	**LPS** (mW)	**Ratio** (%)	**WMS** (um^2)	**LPS** (um^2)	**Ratio** (%)	**WMS** (ps)	**LPS** (ps)	**Ratio** (%)
ARAI	11.7	9.1	-22.22%	31187	28056	-10.04%	885	820	-7.34%
DIF	13.3	12.4	-6.77%	39704	38798	-2.28%	880	883	+0.34%
DIT	15.7	14	-10.83%	46485	44830	-3.56%	880	882	+0.23%
LEE	20	16.1	-19.50%	53886	48892	-9.27%	835	901	+7.90%
MCM	35.9	26.1	-27.30%	87879	76264	-13.22%	838	892	+6.44%
PR	18.4	16	-13.04%	47571	44815	-5.79%	835	835	+0.00%
Average			-16.61%			-7.36%			+1.26%

Power, area, and cycle time comparisons between WMS and LPS (Latency constraint = 1.2x the longest path length)

Experimental Results
– Considerations of Resource Sharing

- LPS_NRS: Minimize total node power (without considering resource sharing)
- LPS_RS: Minimize total function unit power (considering resource sharing)
- ILP: ILP-based approach to directly minimize total FU power (optimal solution)



Actual power consumption

LPS_RS is within 6% of the ILP exact approach and outperforms LPS_NRS by 30%

Conclusions and Future Work

- Conclusions
  - A mathematical programming formulation of the integer time budgeting problem with great flexibility and extensibility
  - Application to low-power scheduling problem
- Future works
  - Apply our ITB formulation to other problems

- Thanks.

Design Closure Driven Delay Relaxation Based on Convex Cost Network Flow [DATE07]

- Problem formulation
  - Design Closure Driven Delay Relaxation problem
  - Essentially a ITB problem with convex objective function
- Solution
  - Transformation to a convex cost integer dual network flow problem
- Comparison with mathematical programming (MP) approach
  - Network flow based algorithm has a better worst-case complexity
  - MP approach allows the utilization of the leading-edge mathematical programming solvers
  - MP approach can be easily extended to support application specific constraints