Behavioral Synthesis of Low Leakage Datapaths

Seminar
University of Tennessee , Knoxville
March 2002

Dr. Srinivas Katkoori
Assistant Professor
Department of Computer Science & Engineering
University of South Florida
katkoori@csee.usf.edu
http://vcapp.csee.usf.edu/~katkoori

Outline of the Talk

- Leakage power in CMOS Circuits
- AUDI Synthesis System – An Overview
- Leakage Power Estimation
  - Library Characterization
- Multi Threshold CMOS (MTCMOS) Technology
- Leakage Power Optimization
  - Leakage Power Management during HLS
  - Area/Performance/Power Tradeoff Analysis
  - Simultaneous Scheduling, Allocation and Mapping for Low Leakage Power
- Conclusions and Ongoing Work

Leakage Power in CMOS Circuits

Leakage Power Consumption in CMOS Circuits

- As devices scale down to Deep Sub-Micron (DSM) regime, leakage power becomes significant
- Scaling down of supply voltage is another cause for increased leakage current
- ITRS projects leakage power to be a "dominant" fraction of power in the near future

"For portable applications, the main issue is now low leakage current, which is absolutely necessary for extended battery life.."

ITRS 2001

Leakage Power Consumption in Microprocessors

- Intel 1 GHz processor (0.18 process)
  - Core power dissipation = 33.0 Watts
  - Static power = 3.74 Watts
- Intel 1.13 GHz processor
  - Core power dissipation = 41.4 Watts
  - Static power = 5.40 Watts

"Total power has increased by only 25% but the static power has increased by 44%."

Intel Corp. Data Sheet

Device Leakage Current (I_{sub})

- Leakage current of a transistor is dependent on:
  - State of the Transistor (ON/OFF)
  - Threshold Voltage (V_t)
  - η – Drain Induced Barrier Voltage
  - n' – Subthreshold swing coefficient
  - γ' – Linearized Body effect coefficient

$$I_{sub} = A.e^{\frac{q}{n'kT}(V_{gs} - V_t - \gamma' V_s + \eta V_{ds})}.(1 - e^{\frac{-qV_{ds}}{kT}})$$

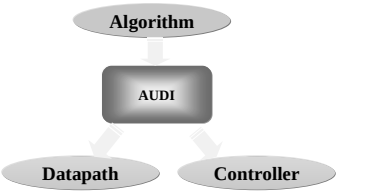
Related Work

- **Gate level Leakage Power Reduction**
  - Halter and Najm (CICC, 1997)
- **Multi-Threshold CMOS (MTCMOS)**
  - Introduction - Mutoh et al (IJSSC, 1995)
  - Sizing issues – Kao and Chandrakasan (IJSSC, 2000)
- **Dual Threshold Voltages**
  - Logic Synthesis – Wei et al (CICC, 2000)
  - During HLS – Khouri and Jha (ICCD, 2000)
  - Delay Analysis – Wang and Vrudhula (ITCAD 2002)
- **Gated-Vdd**
  - High performance low leakage caches - Powell et. al. (ITVLSI 2001)
- **Survey Paper**
  - Roy (ISCAS, 1998)
  - Allam and Elmasry (MSCS, 1999)

Overview of

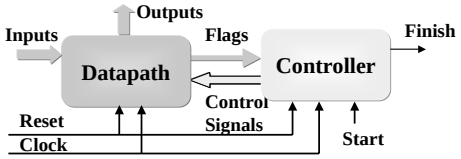
AUDI – A Behavioral Synthesis System

AUtomatic Design Instantiation (AUDI)



- **AUtomatic Design Instantiation (AUDI) from algorithm to architecture**
- **Funded by the National Science Foundation (NSF) under the 2001 CAREER program**

RTL Design Model



- *Reset pulse* initializes the Datapath registers and Control registers
- *Start pulse* signals the design to start processing the inputs
- Datapath and Controller *communicate* with Flags and Control signals
- *Finish* signal indicates that the outputs are valid
- *Clock signal*: ON = Datapath works, OFF = Controller works

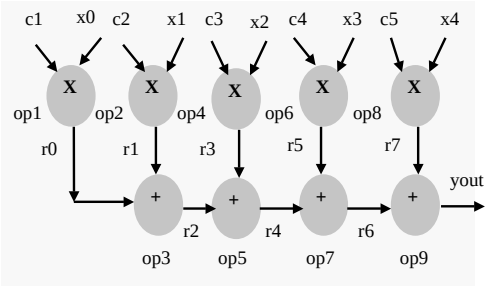
AUDI Synthesis Algorithms

- **Force-directed Scheduling Algorithms**
- **Clique Partitioning Heuristic**
- **0-1 Knapsack Algorithm**
- **Genetic Search Algorithm**
- **Left Edge Algorithm**
- **Vertex Coloring Algorithm**
- **Integer-Linear Programming (ILP) Based**

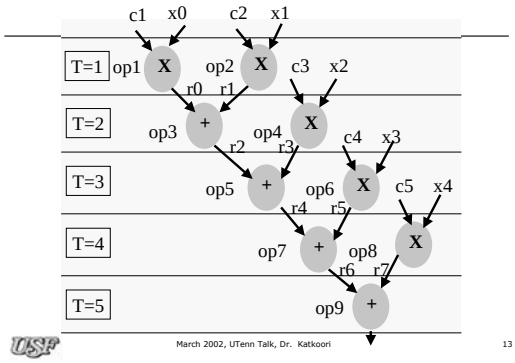
As of April 2002:

- ***Currently Operational***
- ***Under Development***
- ***Near Future (Summer 2002)***

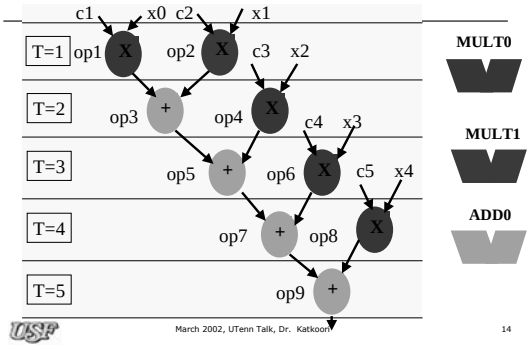
An Illustrative Example FIR Filter



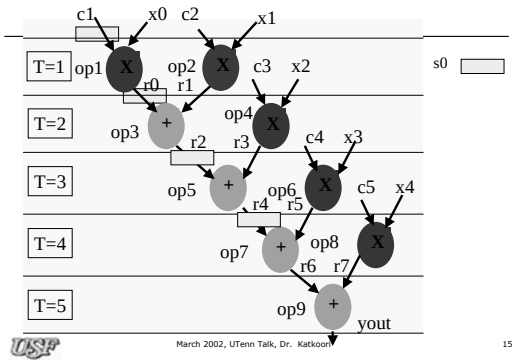
After Scheduling Phase..



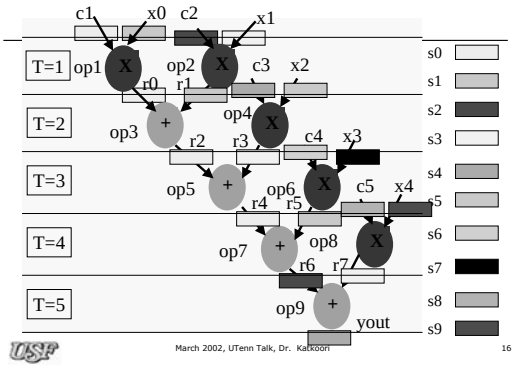
After FU Allocation and Binding..



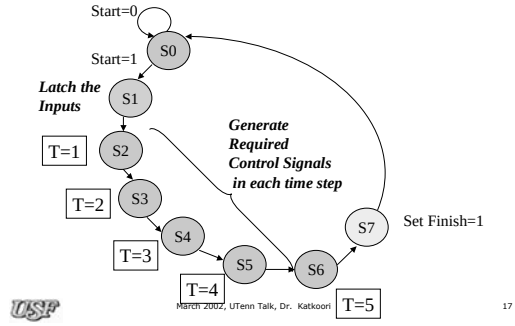
Register Optimization



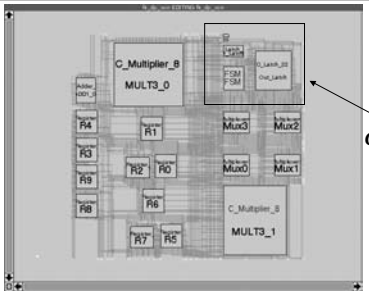
After Register Allocation and Binding..



Controller



FIR Filter: ASIC Layout



Synthesized Designs So Far..

Design Name	Ops	PIs	POs	Edges	Scheds
IIR Filter	5*, 4+	10	1	8	13
FIR Filter	5*, 4+	10	1	8	17
Lattice Filter	5*, 8+	8	3	10	8
FFT4pt	2*, 4+, 4-	6	4	6	25
Elliptic Wave Filter	7*, 26+	9	7	27	5

- CMOS Layouts generated using LagerIV Silicon Compiler (Logic and Layout Synthesis)
- Layouts are validated by HSPICE simulation for 0.35um and 0.18um technology nodes

Synthesized Designs So Far..

Design Name	Operations	PIs	POs	Edges
LMS Filter	9*, 8+	9	4	13
AR Filter	8*, 6-	8	2	12
Acyclic Convolution	6*, 2+, 2-	5	1	9
MA Filter	5*, 4-, 3+	7	1	11
Exponential	6*, 4+	4	1	9

- Validated at the RT-level by VHDL Simulation (CADENCE)
- Used as part of the Digital Circuit Synthesis Class in Spring 2002

RT-Level Leakage Power Estimation

FASL: A Fast Architectural Simulator For Leakage Power

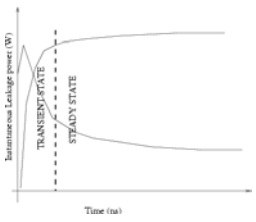
- Estimate leakage power at RT-Level
  - Most existing techniques/tools estimate leakage power at gate-level and lower levels of hierarchy
- Estimate leakage power fast and accurately
  - Leakage Power Estimation using HSPICE is time consuming

Proposed Approach

- Pre-characterize leaf cells for leakage power
- Embed leakage power measuring code in corresponding VHDL models
- On simulation of a hierarchically described VHDL design created using the characterized leaf cells, we obtain leakage power dissipated by the design

Characterization of Leaf cells

- Leakage is statically dependent on current input [Halter and Najm] only after steady state
- During transition, it is dependent on previous input as well
- Accurate leakage characterization during both these periods is done using HSPICE
- Small sizes of leaf cells allow exhaustive characterization using HSPICE



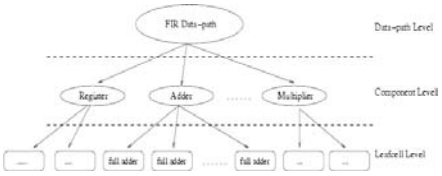
Embedding Leakage measuring code

- Process triggered by inputs to the leafcell
- Leakage energy is accumulated by the following formula :

```
if (T > T_thr) then
        Leak = Leak + T_thr*transient_leakage
else
        Leak = Leak + T_thr*transient_leakage
                + (T – T_thr) * steady_state_leakage
```

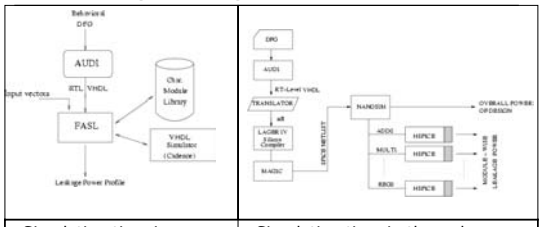
  - where T is the time elapsed since previous trigger and T_{thr} is the threshold time between transient and steady state
  - Transient leakage and steady state leakage are maintained as tables [obtained from characterization]

Hierarchical description in VHDL



- Random logic can be similarly described hierarchically in terms of logic gates (NAND, NOR, INV) as leaf cells

Measuring Leakage Power in Data-paths



• Simulation time in the order of seconds • VHDL Simulator	• Simulation time in the order of minutes/hours • HSPICE

Measuring Leakage Power in Data-paths

	1 Vector			5 Vectors			9 Vectors		
	Hsp	FASL	err%	Hsp	FASL	err%	Hsp	FASL	err%
IIR	3.39	3.37	0.47	15.60	15.57	0.16	27.86	27.82	0.15
FIR	26.12	26.33	0.80	26.65	26.99	1.27	26.74	27.08	1.28
EWF	14.03	13.80	1.68	14.23	13.72	3.56	14.08	13.64	3.13

Simulation Times

	1 Vector		5 Vectors		9 Vectors	
	HSP	FASL	HSP	FASL	HSP	FASL
IIR	2hr47min	1.5s	10hr42min	1.8s	12hr25min	2.3s
FIR	2hr34min	1.5s	12hr36min	1.6s	15hr39min	1.8s
EWF	36min	1.4s	3hr11min	1.5s	7hr39min	1.7s

Measuring Leakage Power in Data-paths ..

- Effect of inclusion of *transient leakage current* on accuracy has been studied

Design	HSPICE (uW)	With Transient: FASL (uW)	With Transient: Error	Without Transient: FASL (uW)	Without Transient: Error
IIR	27.85	27.81	0.15	27.96	0.39
FIR	26.74	27.08	1.28	27.16	1.54
EWF	14.08	13.64	3.13	15.67	11.30

- Effect of inclusion of *transient leakage current* on accuracy has been studied

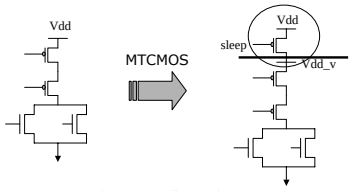
Observations

- Within 5% accuracy of HSPICE
- Simulation times several orders of magnitude lesser than HSPICE
- Very useful for design exploration at high levels of design
- Similar results on random logic benchmarks (MCNC) proves repeatability and wide applicability of simulator

Multi-Threshold CMOS Technology (MTCMOS)

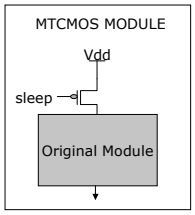
Multi-Threshold CMOS (MTCMOS)

- Sleep transistor introduced
- Controlled by a sleep signal
- Also called Power-gating
- The only leakage current which flows in the IDLE state is through the sleep transistor which is OFF



MTCMOS Module Library

- MTCMOS component library
  - functional units
  - storage units
  - interconnect units,
- Can be powered ON/OFF using sleep control signals
- Controller generates appropriate sleep control signals to switch idle modules OFF
- Each module has been characterized with respect to area/power/performance

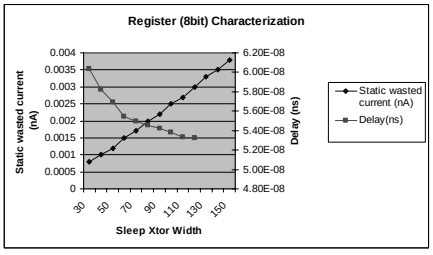


Multi-Threshold CMOS Module Library Characterization

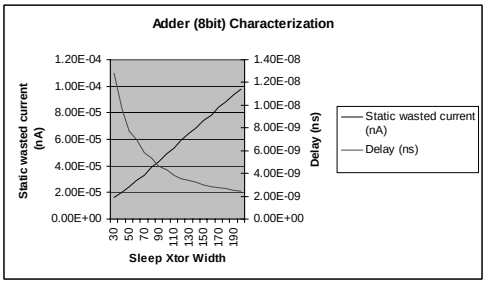
Module Library Characterization

- Delay introduced by Sleep Transistor $\propto$ 1/(Area)
- Static Wasted Current $\propto$ Area
- Therefore, Delay and Power are conflicting with each other
- Need to find optimal sleep transistor width for
  - Low leakage power
  - High Performance
- For each module,
  - Generate Layout using Lager IV Silicon Compiler
  - Leakage Power: Simulate using *nanosim* to measure leakage power using long random input sequence
  - Delay: Measure the propagation delay using *PathMill*

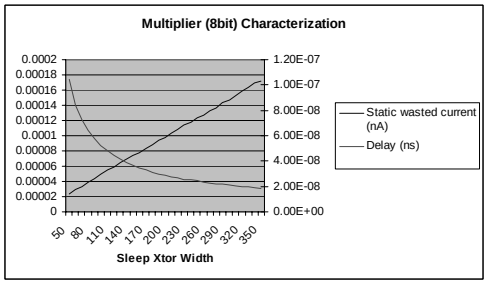
8-bit Register



8-bit Ripple Carry Adder



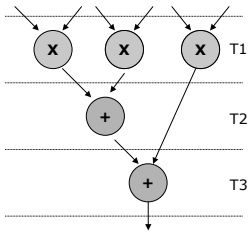
8-bit Array Multiplier



Leakage Power Management Using MTCMOS Technology

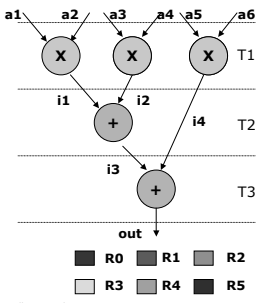
Motivation for Leakage Power Management during HLS

- Dataflow graph implemented with 3 multipliers and 2 adders
- ASAP schedule
- Multipliers can be switched OFF in time steps T2 and T3
- Adder can be switched OFF in time step T1



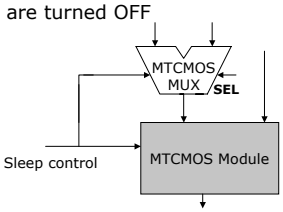
Motivational Example contd.

- Switching OFF registers
- A possible register allocation/binding is shown
- 6 Registers have been allocated
- Registers R2, R3, and R4 can be turned OFF in T2 and T3

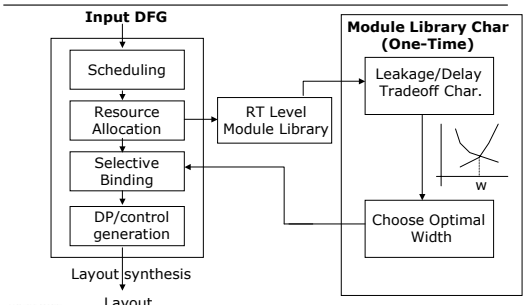


Interconnect Units

- Interconnect units (Mux) can be powered OFF during the time-steps when the module they feed are turned OFF



Proposed Approach



Register Allocation/Binding for Low leakage Power

- Leakage power can be minimized if registers are allocated and mapped such that their idle times are maximized
- Algorithm based on Clique partitioning Heuristic proposed by Tseng and Siewiorek(1983)
- Added cost function =

$$W_{sleep}.(\text{sleep time}) + W_{trans}.(\text{transitions})$$

- sleep time cost is the number of time steps the register would be in IDLE mode if the current edge is mapped to the register
- transitions cost is the number of transitions between IDLE and ACTIVE state the register would make if the current edge is mapped to the register

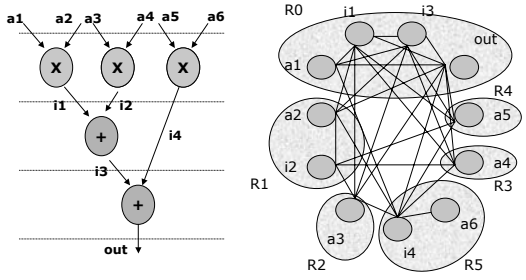
Cost modifications

- Increasing sleep time increases the possibility of saving more leakage energy
- Reducing the number of transitions reduces the possible short circuit power dissipation due to transitions between idle and active states
- W_{sleep} and W_{trans} are user defined constants

Clique Partitioning Heuristic

1. Input is an undirected graph
2. Each node is a carrier in the DFG and edges exist if the carriers are compatible
3. Cliques are formed by merging nodes in each iteration
4. Merging is done by
   1. Find all neighbors (Y) of node to be merged
   2. Find all incompatible nodes and which exclude the minimum number of nodes in Y
   3. Choose node with highest leakage cost factor
5. Assign merged node higher priority and repeat until all nodes are merged

Clique partitioning contd.

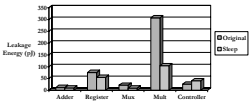


Experimental Results - I

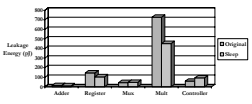
- Design Examples : FIR/IIR/Lattice filters
- Schedule : ASAP
- Simulations : Avant! HSPICE
- Technology Details: Vt=0.3V, T= 25°C, 0.35um

Design	Operations	PIs	POs
IIR	5 Mult, 4 Add	10	1
FIR	5 Mult, 4 Add	10	1
Lattice	5 Mult, 8 Add	8	1

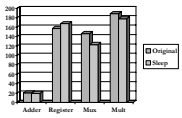
Leakage Energy Savings



IIR Filter (57.26%)



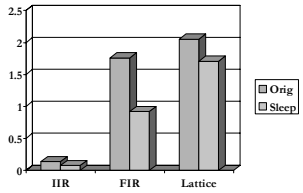
FIR Filter (33.60%)



Lattice Filter (4.88%)

Overall Power Savings

- By gating the supply voltage during inactive periods, we also obtained a reduction in overall power consumption
- Average dynamic power reduction = 34.20%



LSAM: A Simultaneous Scheduling Allocation and Binding Algorithm for Leakage Power Optimization

LSAM

- Simultaneous Scheduling, Allocation, and Mapping for low leakage power
- Force Directed Approach
- Algorithm is a modified version of SAM proposed by Cloutier and Thomas
- Minimizes leakage power by maximizing the sleep time of various modules during scheduling/allocation/mapping
- Global solution

Modified forces

- For each possible mapping of an operation to a specific time step, a force is introduced which factors in the effect of that mapping on mapping its input/output edges to a register (reg_cost)
- For each possible operation/register mapping to a time step, there is a possible multiplexer introduced (mux_cost)

New costs

- Register cost =
  - Reg_instance_cost → Cost of adding a new register
  - + Sleep_time_cost → time the register spends in idle time
  - + Transition cost → number of times the register transitions between idle and active states
- Mux cost =
  - New_mux_cost → Cost of adding a new multiplexers
  - + Mux_complexity_cost → Cost of adding a 4-1 Mux > Cost of adding a 3-1 Mux > Cost of adding a 2-1 Mux

Register Allocation Table Snapshot

- IIR Filter
  - PIs : a1, y1, a2, y2, b0, x0, b1, x1, b2, x2
  - Registers : r0 – r9
  - Idle periods

Time /Inst	r0	r1	r2	r3	r4	r5	r6	r7	r8	r9
T0	a1	y1	a2	y2	b0	x0	b1	x1	b2	X2
T1			i3	i4	i2					
T2	i1		i5	i7						
T3	i6				I8					
T4		yout								

LSAM contd

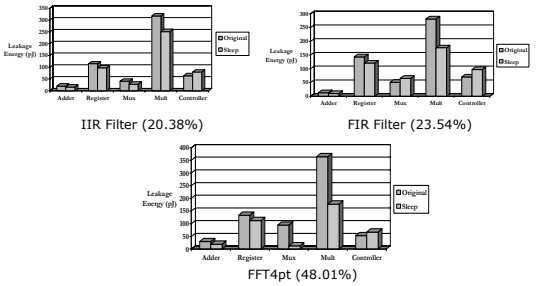
- Registers are allocated in a allocate-when-needed basis → area minimal
- Complexity is $O(cn^2ir)$
- Scheduling and mapping is performed taking sleep times into account
- Run times are very comparable to SAM algorithm
- Synthesizes designs with much lesser leakage power

Experimental Results - II

- Design Examples : FIR/IIR/FFT4pt filters
- Schedules : SAM and LSAM
- Simulations : HSPICE

Design	Operations	PIs	POs
IIR	5 Mult, 4 Add	10	1
FIR	5 Mult, 4 Add	10	1
FFT4pt	2 Mult, 2 Add, 4 Sub	4	4

Leakage Energy Savings - II



IIR Filter (20.38%) FIR Filter (23.54%)

FFT4pt (48.01%)

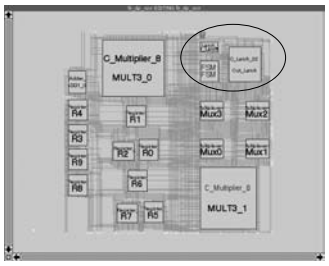
Area/Performance/Power Tradeoff Analysis

Overhead Analysis

- Controller Overheads
- Area Overhead
  - Knapsack formulation
- Power Overhead
- Performance Recovery

Controller Overhead

- Controller in the sleep version is larger in area than the original controller and dissipates more power
- Area/power of controller is less than 20% of the total area/power of design



Area Overhead due to MTCMOS

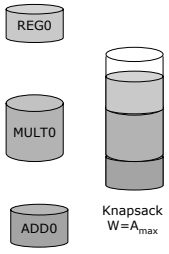
- Area
  - The bulky sleep transistors impose an area overhead
  - Average area overhead in the designs considered was 27.69%
- Area and power overhead analysis is formulated as a 0-1 knapsack problem
  - Designer has handle on maximum permissible (sleep) area overhead

0-1 Knapsack Problem

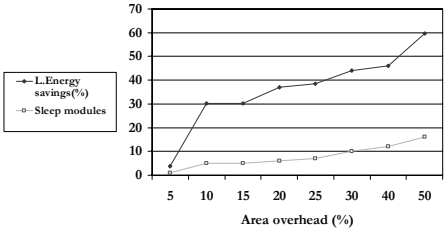
- Problem
  - Given:
    - Set of N unique items, each item with associated value v_i and weight w_i
    - a knapsack of capacity W
  - Choose a Subset of items such that:
    - The total weight of chosen items, $\Sigma w_i < W$
    - Maximize Σv_i
- Solution:
    - Dynamic Programming Based Algorithm which makes a decision for each item to be chosen or not i.e., a 0-1 decision problem

Area/Power Tradeoff 0-1 Knapsack Formulation

- Designer chooses maximum permissible sleep area (% overhead) A_{max} (=W)
- *N* RT-Level instances for each of which
  - v_i = Leakage power savings
  - w_i = optimal sleep transistor width
- Choose a subset of modules that will be implemented in MTCMOS technology while satisfying area overhead
- Maximize Overall Leakage Power Savings



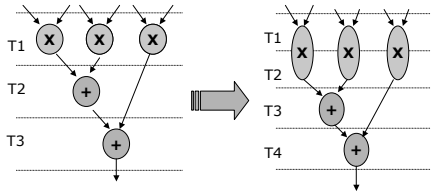
IIR Filter



Performance Recovery

- Introducing MTCMOS modules in the data-path decreases performance
- Delay overhead introduced due to MTCMOS modules, is dependent on the complexity of the module
  - delay $_{MTCMOS\ multiplier}$ > delay $_{MTCMOS\ register}$
- We observed that the delay overhead was mainly due to the MTCMOS multiplier
- Performance is recovered through Multi-cycling and Introduction of slack

Performance Recovery contd.



- Multipliers multi-cycled over two clock cycles and introduction of slack of one clock cycle

Ongoing Work

- Overhead analysis to optimize performance by choosing optimum sleep transistor widths
- Incorporating selective binding into LSAM
- Exploring architectural transformations during behavioral synthesis targeting lower leakage power

Conclusions

- Leakage power management techniques
- A Fast Architectural Leakage Power Simulator for Hierarchical VHDL descriptions
- A modified clique partitioning heuristic for low leakage register allocation/binding
- Leakage-aware Simultaneous Scheduling, Allocation and Mapping Algorithm
- First step in the automated synthesis of low leakage power portable systems

Acknowledgements

National Science Foundation (NSF)

Thank You!!