Design of an Efficient Power Delivery Network in an SoC to Enable Dynamic Power Management

Behnam Amelifard
Massoud Pedram

Department of Electrical Engineering
University of Southern California

Outline

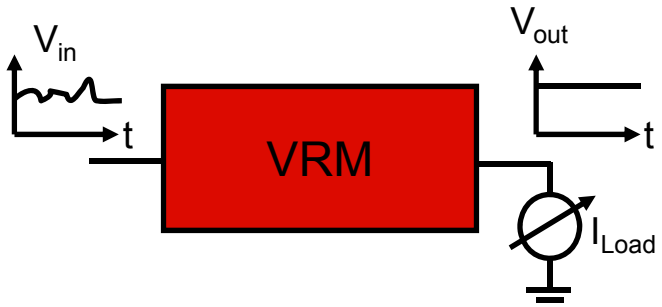
- Introduction
- VRM Allocation Supporting DVS
  - Conventional Technique
  - Proposed Technique
- Experimental Results
- Conclusions

Introduction

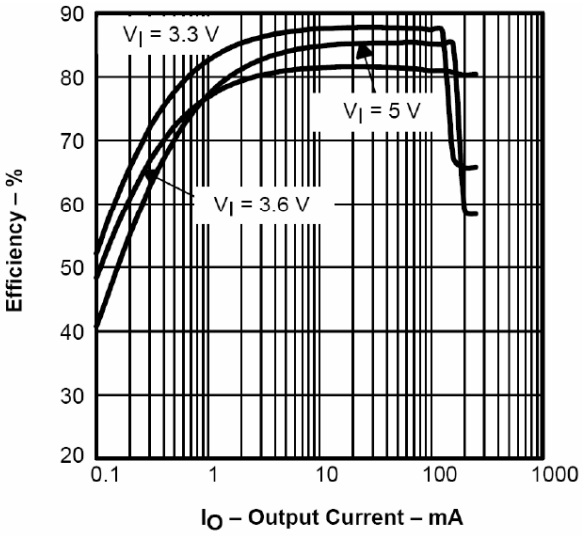
- Power delivery network (PDN) is a critical component of a state-of-the-art design
- PDN design comprises of three steps
  - Establishing a PDN target impedance
    - Target impedance should be met over a broad frequency range
  - Designing system-level decoupling network
    - Decaps act as charge reservoirs
  - Allocating voltage regulator modules (VRM's)
    - VRM provides constant DC output voltage

Voltage Regulator Modules (VRM's)

- VRM tasks
  - Voltage regulation
  - DC-DC conversion



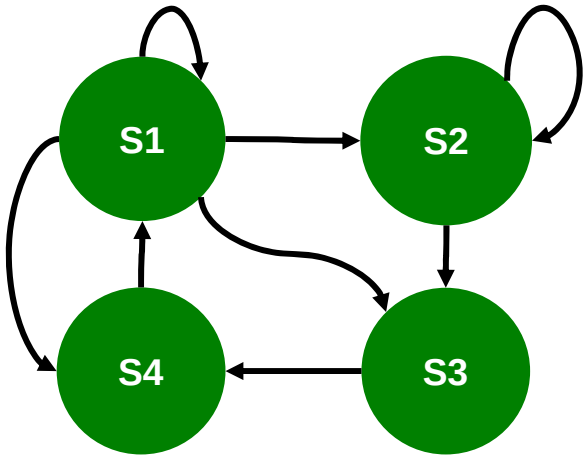
- Power efficiency



- Different types of VRM's
  - Inductor-based VRM's
  - Charge-pump VRM's
  - Linear VRM's

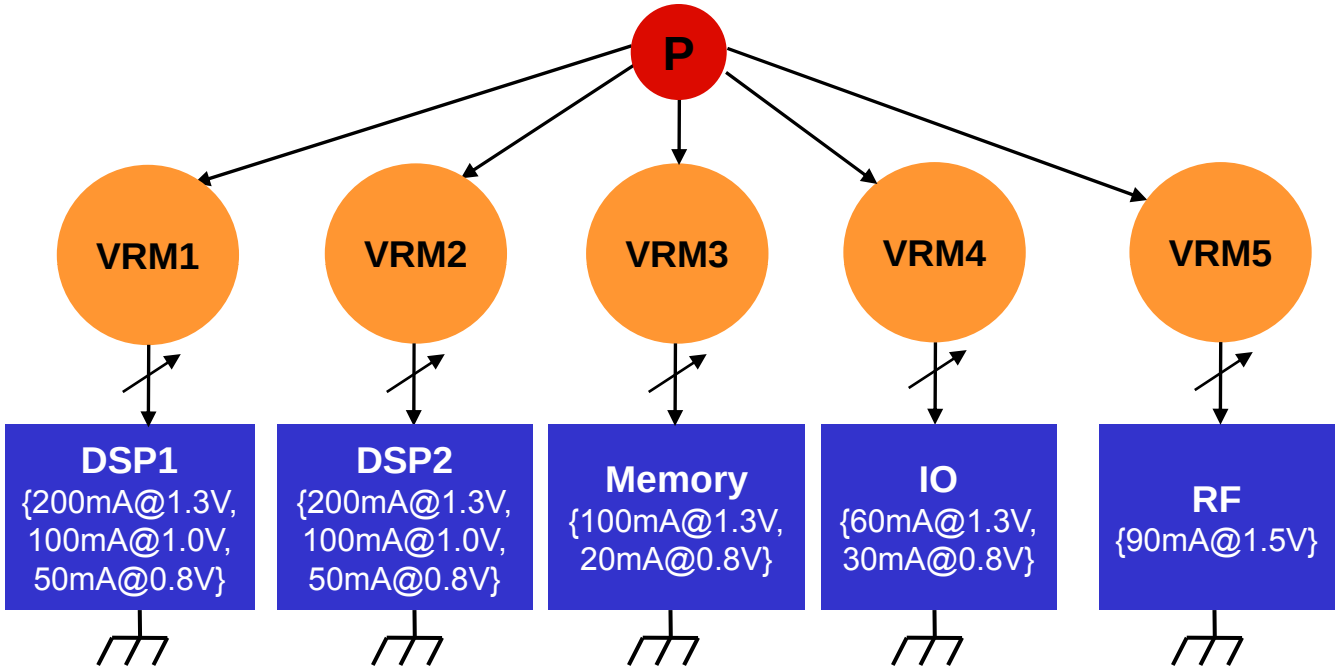
Dynamic Voltage Scaling

- DVS: Dynamically adjust supply voltages of functional blocks (FB's) in a system-on-chip (SoC)
  - Objective: minimize power dissipation while meeting performance demands
- Power manager decides when to switch SoC's *"power-performance state"* (*PPS*)
  - Each PPS corresponds to a particular combination of voltage level assignments to various FB's



VRM Allocation Supporting DVS

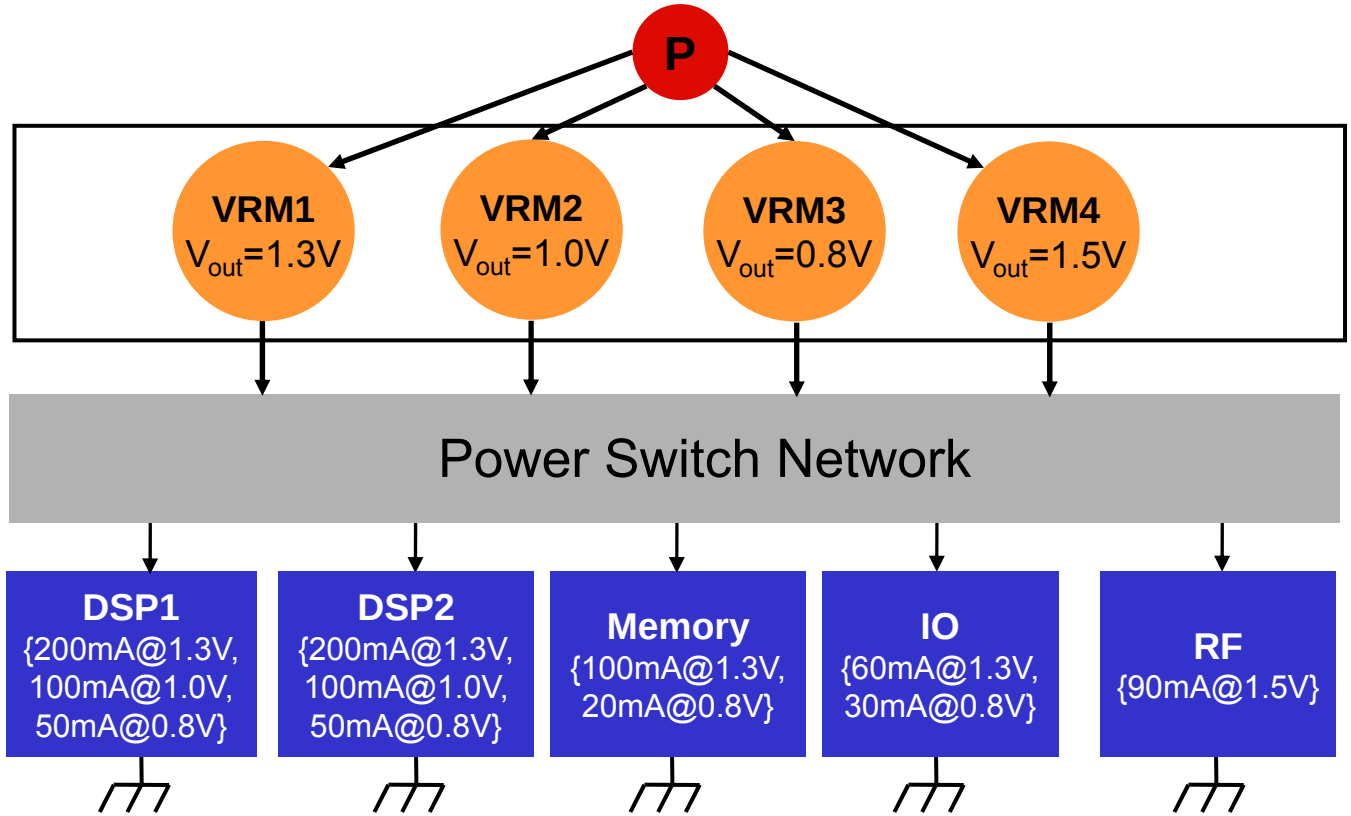
- Conventional technique: each FB has its own variable-Vout VRM (a.k.a. *dynamic VRM*)



- Drawbacks:
  - Number of VRM's equals number of FB's
  - Design of dynamic VRM is more challenging and its cost is higher than that of a static VRM
  - Efficiency of static VRM is optimized for a specific Vout; efficiency of dynamic VRM varies as a function of chosen Vout

Proposed Technique

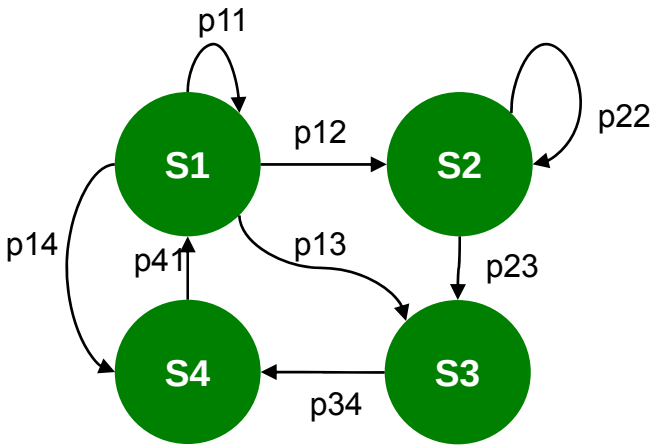
- The proposed PDN comprises of two parts:
  - A Power Conversion Network (PCN)
    - Generate all voltage levels needed by FB's
  - A Power Switch Network (PSN)
    - Dynamically connects FB's to appropriate VRM's in the PCN



Markov Model of the System

- PPS transitions modeled by a stationary time-independent transition matrix [p_{ij}]
- In each state of Markov chain, supply voltage level of all FB's specified.
- Probability of being in state *i* is a constant value π_i

$$\pi_i = \sum_{j \in S} \pi_j p_{ji}$$



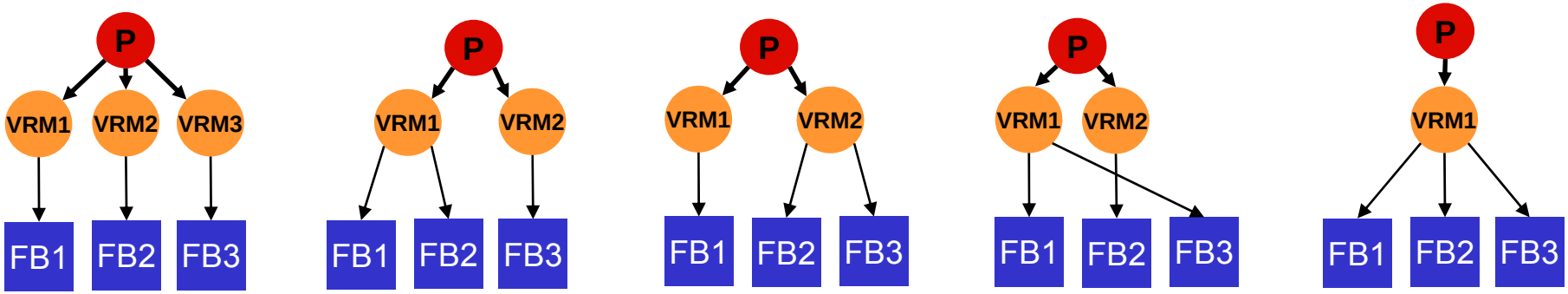
PCN Optimization: Problem Definition

- PCN Optimization supporting DVS (PCODS) Problem:
  - Given is:
    - A library $\mathcal{R}$ of VRM's; $\forall r \in \mathcal{R}$:
      - V_{out}, min and max V_{in}, max I_{out}
      - Efficiency $\eta_r = f(V_{in}, I_{out})$
      - Associated cost c_r
    - A set $\mathcal{F}$ of FB's; $\forall l \in \mathcal{L}$: $\{(V_l, I_l)\}$
    - A power source P, with nominal voltage V_P
    - A Markov chain model $\mathcal{M}$ of system:
      - In each state of Markov chain, the supply voltage level of FB's specified.
  - Objective is
    - Build a network of VRM's b/w P and FB's to minimize

$$V_P I_P + \lambda \sum_{r \in PCN} c_r$$

PCN Optimization

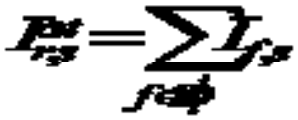
- Definition: *Voltage domain* $\mathcal{D}(V_i)$ is the set of all FB's that require voltage level V_i in one of their PPS's
- Different options to deliver power to FB's in a voltage domain

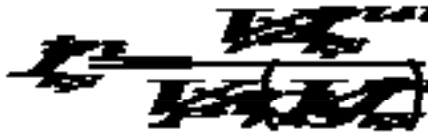


- *Partition* of set $\mathcal{D}_i$: a collection of disjoint subsets, $\{\mathcal{D}_i^j\}$, whose union is $\mathcal{D}_i$
  - Each of subsets $\mathcal{D}_i^j$ is a *part*
- Bell numbers: count the number of set partitions for a set of *n* elements

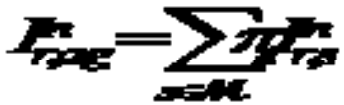
Power Consumption of a Fixed VRM Tree

- In state *s*, output current of VRM delivering power to $\mathcal{D}_i^j \subseteq \mathcal{D}_i$
- VRM input current





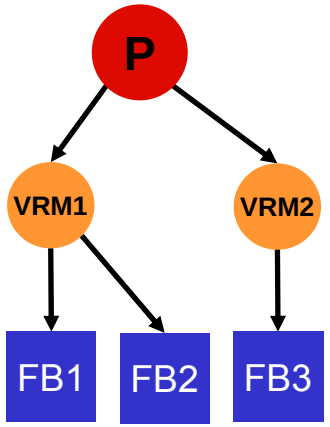
- Average VRM input current



- Average current drawn by $\mathcal{D}_i$ and cost of VRM's

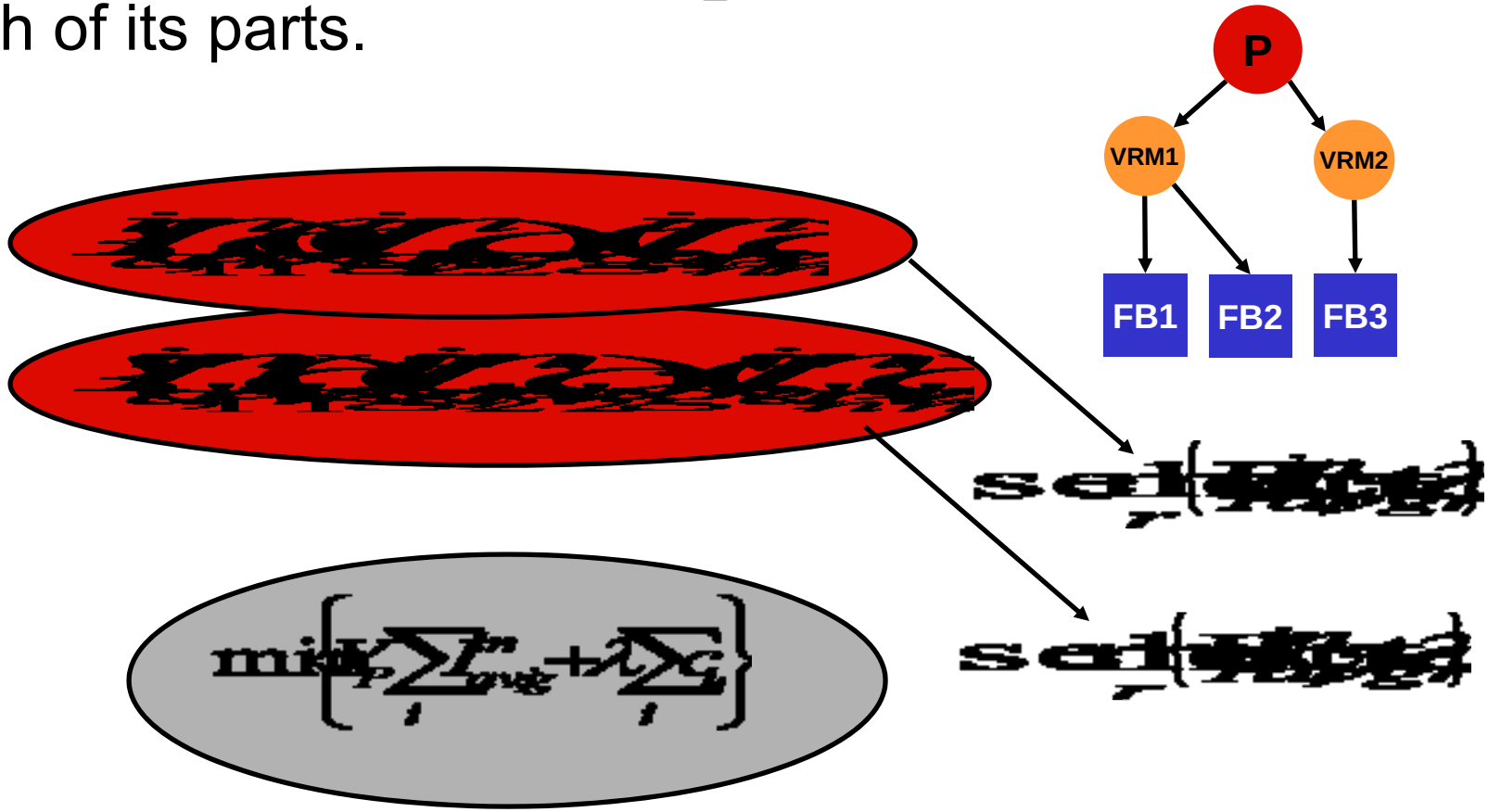
,

- Weighted sum of power and cost



Best VRM Selection

- Lemma: A VRM assignment to a partition of $\mathcal{D}_i$ is optimum, if and only if, $V_{P_{i,avg}}^{T} + \lambda c_i$ is minimized in each of its parts.



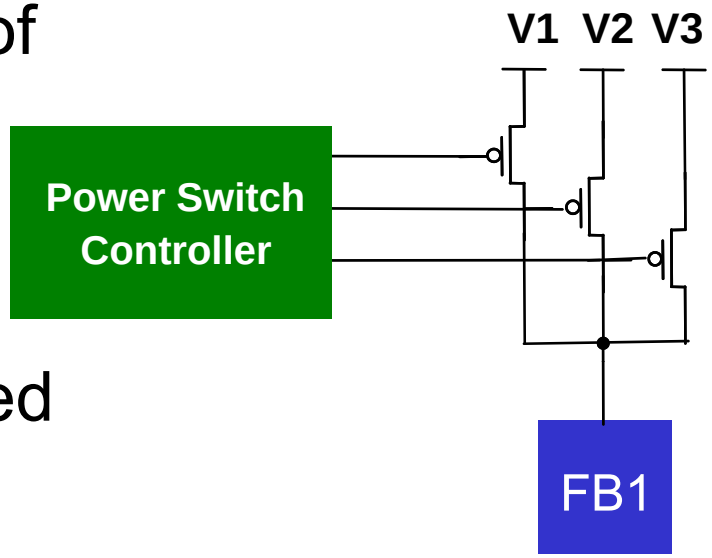
PCN Optimization: optPCN Algorithm

Algorithm *optPCN*($\mathcal{R}$, $\mathcal{L}$, $\mathcal{M}$, V_P)
Begin
  For each voltage domain $\mathcal{D}_i$
    $optCost = \infty$; $optVRM = \{\}$
    For each set partition of $\mathcal{D}_i$ such as $\{\mathcal{D}_i^1, \ldots, \mathcal{D}_i^n\}$
      For each part $\mathcal{D}_i^j$
        Select the best VRM r that minimizes
      End
      $newCost = \Sigma_j cost_j$
      If ($newCost < optCost$)
        $optCost = newCost$
        $optVRM = \{VRM[j]\}$
      End
  End
  Return ($optCost$, $optVRM$)
End

- Worst case running time of algorithm: $O\left(|\mathcal{R}| \cdot |\mathcal{M}| \cdot |\mathcal{F}| \cdot B_{|\mathcal{F}|+1}\right)$

PSN Optimization

- PSN switches supply voltage of FB's when a new PPS is commanded by the power manager



- Power switches should be sized carefully to avoid IR-drop
  - From the alpha-power model

$$I_{DS} = k\frac{W}{L_{eff}}\left(\frac{V_{GS}-V_{th}}{V_{dd}-V_{th}}\right)^{\alpha/2} V_{DS}$$

  - For an allowable voltage drop $\Delta V_{f,v}$

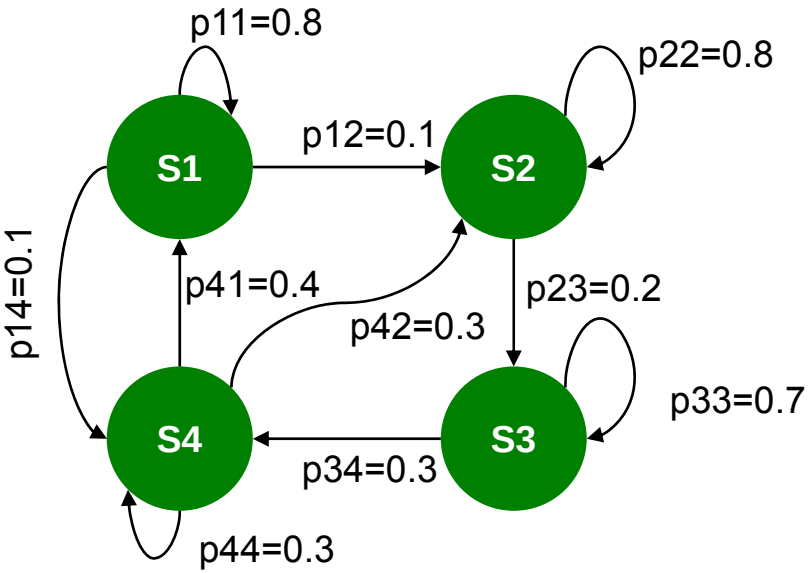
$$W_{f,v}^{\min} = \frac{I_{f,v}L_{eff}}{k\Delta V_{f,v}}$$

Experimental Setup

- Algorithm is implemented in C++
- A set of 30 commercial VRM's from Texas Instruments and National Semiconductors are used to build a library of VRM's
  - VRM efficiency modeled as a PWL function
  - VRM cost assumed as the dollar cost for a 1000-unit purchase
- A set of test-benches (TB) used for evaluating proposed technique
  - TB1 specs

DSP1:{200mA@1.3V, 100mA@1.0V, 50mA@0.8V}
DSP1:{200mA@1.3V, 100mA@1.0V, 50mA@0.8V}
MEM:{100mA@1.3V, 20mA@0.6V}
IO:{60mA@1.3V, 30mA@0.8V}
RF:{90mA@1.5V}

S1: {V_{DSP1}=1.3, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5}
S2: {V_{DSP1}=1.0, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5}
S3: {V_{DSP1}=0.8, V_{DSP2}=1.0, V_{MEM}=1.3, V_{IO}=0.8, V_{RF}=1.5}
S4: {V_{DSP1}=0.8, V_{DSP2}=0.8, V_{MEM}=0.8, V_{IO}=0.8, V_{RF}=1.5}



Experimental Results

- Minimizing power consumption (λ=0)

Circuit	No. FB	No. States	PDN Power Reduction (%)	PDN Cost Reduction (%)	Runtime (sec)
C1	5	4	38.5	1.1	<1
C2	6	4	40.4	5.0	<1
C3	8	5	34.2	-2.8	<1
C4	10	10	30.1	29.7	13
C5	12	10	27.9	8.1	70

- Trading off power for cost
  - Choose λ such that PDN power loss does not increase more than 10% of its optimal value

Circuit	PDN Power Increase over Power-optimal Solution (%)	PDN Cost Reduction over Power- optimal Solution (%)
C1	10.0	53.0
C2	4.3	46.9
C3	8.9	57.9
C4	9.6	26.1
C5	10.0	52.9

Summary

- Presented a new technique and an accompanying PDN architecture to realize an efficient PDN for SoCs with DVS capability
  - The proposed PDN is composed of two parts: PCN and PSN
- Described an algorithm to optimally select the best VRM's in the PCN
- The proposed solution reduces power loss of PDN by 34% while reducing its cost by 8%