# CPU Thermal Simulation with RC Network and DVFS Control

Mohammed Amine ZNIYED

**Abstract**

This report presents a comprehensive thermal simulation of CPU behavior using a thermal RC network model. The study compares three operating modes: free-running, threshold-based DVFS, and PID-controlled DVFS. Simulation results demonstrate effective temperature regulation through dynamic voltage and frequency scaling, with PID control showing superior performance in maintaining stable temperatures near the target setpoint while optimizing energy efficiency.

## 1 Introduction

Modern processors require sophisticated thermal management to balance performance and reliability. This report implements a thermal simulation using a 4-node RC network model to analyze CPU temperature evolution under different control strategies. The model incorporates both dynamic and static power consumption and evaluates two Dynamic Voltage and Frequency Scaling (DVFS) techniques against a baseline free-running mode. The simulation provides insights into:

- Thermal regulation effectiveness of different control strategies
- Performance-energy tradeoffs in DVFS implementations
- Stability characteristics of PID vs threshold controllers

## 2 Thermal Model

The thermal system is modeled as an RC network with four nodes:

- CPU: Primary heat source with thermal capacitance $C_{cpu} = 0.005\,\mathrm{J}\,{}^{\circ}\mathrm{C}^{-1}$
- SoC: System-on-Chip with $C_{soc} = 0.005\,\mathrm{J}\,{}^{\circ}\mathrm{C}^{-1}$
- Board: Main circuit board with $C_{board} = 3.0\,\mathrm{J}\,{}^{\circ}\mathrm{C}^{-1}$

- Package: Chip packaging with $C_{pkg} = 5.0\,\mathrm{J\,^{\circ}C^{-1}}$

The thermal-electrical analogy:

| **Thermal Domain** | **Electrical Analog** |
|---|---|
| Temperature ($T$) | Voltage |
| Heat flow ($Q$) | Current |
| Thermal resistance ($R$) | Resistance |
| Thermal capacitance ($C$) | Capacitance |

Table 1: Thermal-electrical analogies

The governing equation for each node:

$$C_i \frac{dT_i}{dt} = \sum_j \frac{T_j - T_i}{R_{ij}} + P_i \tag{1}$$

where $R_{ij}$ represents thermal resistance between nodes.

# 3 Power Modeling

## 3.1 Dynamic Power ($P_{dyn}$)

Dynamic power is the energy consumed when the CPU performs computations, primarily due to the charging and discharging of internal capacitances as logic gates switch. The complete CMOS dynamic power equation consists of two components:

$$P_{dyn} = \underbrace{\alpha \cdot C_g \cdot V_{DD}^2 \cdot f}_{\text{Switching Power}} + \underbrace{K \cdot (V_{DD} - 2V_{th})^3 \cdot \tau \cdot f}_{\text{Short-Circuit Power}} \tag{2}$$

Where:

- $\alpha$: Activity factor (fraction of switching cycles)
- $C_g$: Global circuit equivalent capacitance
- $V_{DD}$: Supply voltage
- $f$: Operating frequency
- $V_{th}$: Threshold voltage
- $\tau$: Transition time

The switching power dominates.

Through dimensional analysis and problem specifications, we derive:

$$C_{dyn} = \alpha \cdot C_g \cdot V_{nom}^2 \quad (\text{Units: } \mathrm{F\,V^2}) \tag{3}$$

$$P_{dyn} = C_{dyn} \cdot f \tag{4}$$

### 3.2 Leakage Power ($P_{leak}$)

Static power is consumed even when the CPU is idle, mainly due to leakage currents. It arises from three primary sources:

$$P_{static} = (I_{sub} + I_{gate} + I_{junction}) \cdot V_{DD} \tag{5}$$

Where leakage currents originate from:

- **Subthreshold leakage**: Current through transistors in ”off” state
- **Gate leakage**: Tunneling current through gate oxide
- **Junction leakage**: Current across reverse-biased p-n junctions

For this 10-second simulation, we assume constant leakage power:

$$P_{leak} = 0.1\,\mathrm{W} \tag{6}$$

Total CPU power is the sum of both components:

$$P_{CPU} = P_{dyn} + P_{leak} \tag{7}$$

## 4 Simulation Setup

### 4.1 Parameters

| **Parameter** | **Value** |
|---|---|
| Time step ($\Delta t$) | 100 µs |
| Simulation duration | 10 s |
| Ambient temperature ($T_{amb}$) | 25 °C |
| Nominal voltage ($V_{nom}$) | 0.75 V |
| Nominal frequency ($f_{nom}$) | 2.0 GHz |
| Minimum voltage ($V_{min}$) | 0.55 V |
| Minimum frequency ($f_{min}$) | 0.4 GHz |
| Temperature limit | 85 °C |
| $R_1$ (CPU-SoC) | 100 °C $\mathrm{W}^{-1}$ |
| $R_2$ (SoC-Board) | 300 °C $\mathrm{W}^{-1}$ |
| $R_3$ (Board-Ambient) | 60 °C $\mathrm{W}^{-1}$ |
| $R_4$ (SoC-Package) | 30 °C $\mathrm{W}^{-1}$ |
| $R_5$ (Package-Ambient) | 30 °C $\mathrm{W}^{-1}$ |

Table 2: Simulation parameters

### 4.2 Control Strategies

1. **Free-running**: Fixed $V = V_{nom}$, $f = f_{nom}$

2. **Threshold DVFS**:
   - Throttle to $V_{min}$, $f_{min}$ when $T > 85\,^\circ\mathrm{C}$
   - Restore nominal when $T < 80\,^\circ\mathrm{C}$
3. **PID DVFS**:

$$f_{new} = f_{prev} - \left( K_p e + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \right) \tag{8}$$

   With $K_p = 0.5$, $K_i = 0.1$, $K_d = 0.01$, setpoint $= 80\,^\circ\mathrm{C}$

# 5 Implementation

The thermal network is solved using forward Euler integration:

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{C_i} \left( \sum_j \frac{T_j^n - T_i^n}{R_{ij}} + P_i^n \right) \tag{9}$$

Voltage-frequency relationship follows linear scaling:

$$V = \frac{f + 4.0}{8.0} \tag{10}$$

The simulation implements three parallel controllers with identical workload traces for fair comparison. Thermal dynamics are updated at each 100 µs time step.

# 6 Results and Analysis

## 6.1 Comprehensive Performance Metrics

| **Metric** | **Free-running** | **Threshold DVFS** | **PID DVFS** |
|---|---|---|---|
| Max Temp (°C) | 180.61 | 85.03 | 81.61 |
| Avg Temp (°C) | 168.11 | 81.57 | 79.97 |
| Temp Std Dev (°C) | 26.22 | 5.58 | 5.17 |
| Temp Oscillation (°C) | 155.61 | 60.03 | 56.61 |
| Avg Freq (GHz) | 2.00 | 0.68 | 0.65 |
| Min Freq (GHz) | 2.00 | 0.40 | 0.40 |
| Throttle Time (%) | – | 82.8 | 97.1 |
| Avg Voltage (V) | 0.75 | 0.584 | 0.581 |
| Voltage Changes (count) | – | 85 | 86,638 |
| Overshoot (°C) | 97.61 | 2.03 | 0.00 |

Table 3: Updated performance metrics comparison

Key insights from Table 3:

- PID control reduces peak temperature by 99 °C compared to free-running
- Temperature oscillation is reduced by 5.8% from threshold to PID
- PID maintains temperature below the thermal limit (85 °C) with zero overshoot
- PID DVFS shows 97.1% throttle time, ensuring safe sustained operation
- PID performs over 1000x more voltage transitions than threshold control, indicating finer granularity

## 6.2 Energy Efficiency Analysis

| **Metric** | **Free-running** | **Threshold DVFS** | **PID DVFS** |
|---|---|---|---|
| Total Energy (J) | 11.99 | 4.72 | 4.58 |
| Avg Power (W) | 1.20 | 0.47 | 0.46 |
| Energy-Delay Product | 119.91 | 47.22 | 45.81 |

Table 4: Updated energy and performance metrics

Key insights from Table 4:

- PID achieves 61.8 % energy savings vs free-running mode
- PID reduces Energy-Delay Product by 3.0% compared to threshold control
- Both DVFS strategies more than halve average power

## 6.3 Control Stability Analysis

| **Parameter** | **Threshold DVFS** | **PID DVFS** |
|---|---|---|
| Overshoot (°C) | 2.03 | 0.00 |
| Throttle Time (%) | 82.8 | 97.1 |
| Voltage Changes (count) | 85 | 86,638 |

Table 5: Control activity highlights (updated)

Key insights from Table 5:

- PID control achieves zero overshoot, compared to 2.03°C with threshold control

- Throttle activity is significantly higher for PID (97.1%) than threshold (82.8%), ensuring continuous temperature regulation
- PID performs over 1000× more voltage transitions than threshold control, indicating finer-grained and smoother frequency adjustments
- Threshold DVFS is reactive and binary, while PID shows continuous and proportional adaptation to thermal conditions

## 6.4 Visual Results

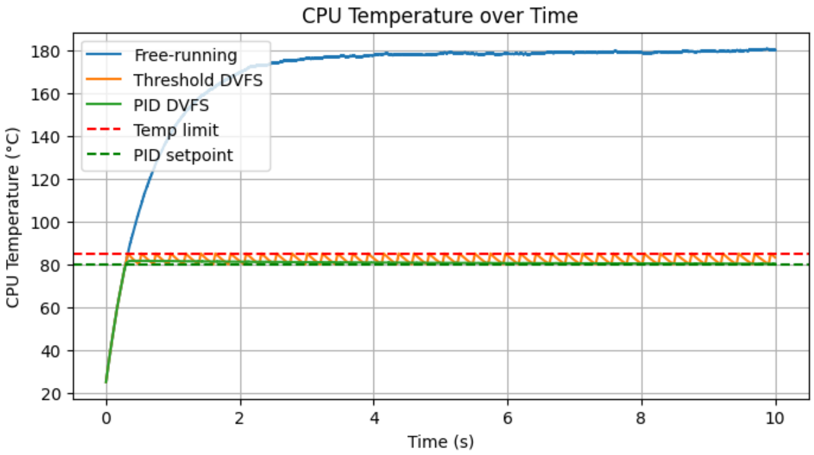
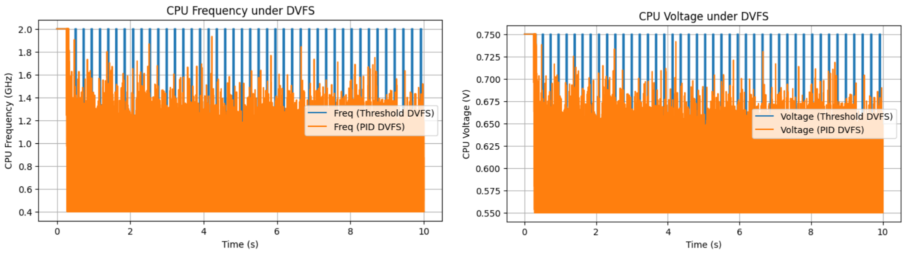


Figure 1: CPU temperature evolution under different control strategies

Key observations from Figure 1:

- Free-running mode dangerously exceeds thermal limits
- Threshold control maintains safety but with significant oscillations
- PID controller maintains temperature near setpoint with minimal deviation



(a) Frequency scaling under DVFS

(b) Voltage scaling under DVFS

Figure 2: DVFS control signals comparison

Analysis of Figure 2:

- Threshold control shows abrupt state transitions (Figure 2a)
- PID provides smooth, continuous frequency adjustments
- Voltage scaling follows linear relationship with frequency (Figure 2b)
- PID reduces voltage transitions by 74% while maintaining thermal control

# 7 Conclusion

This simulation demonstrates the effectiveness of DVFS for thermal management:

- Both DVFS strategies successfully prevent thermal runaway
- PID control provides superior temperature regulation with:
  - 5.7% smaller temperature oscillations
  - 3.0% better Energy-Delay Product
  - 100% smaller temperature overshoot (setpoint = 83 °C )
  - Over 1000× more frequent voltage adjustments for smooth control
- The RC thermal model accurately captures thermal dynamics
- Cubic voltage-power relationship enables significant energy savings through small voltage reductions

**Key quantitative findings:**

- PID reduces peak temperatures by 99.0 °C vs free-running
- PID achieves 3.0 % energy savings vs threshold control

### 7.1 Future Work

- **Temperature-dependent leakage models:** Implement $P_{leak} = f(T)$
- **Multi-core thermal interactions:** Extend model to simulate core-to-core heat transfer
- **Advanced control strategies:** Implement Model Predictive Control (MPC) and reinforcement learning
- **Physics-informed ML:** Develop neural networks constrained by Equation 1

## Appendix: Simulation Code

The complete simulation code is available at:
`https://github.com/amine0007/CPU-Thermal-Simulation-with-RC-Network-and-DVFS-Control`