SUIT

Secure Undervolting with Instruction Traps

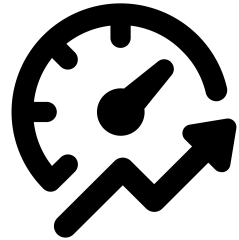
Jonas Juffinger, Stepan Kalinin, Daniel Gruss, Frank Mueller

ASPLOS 2024, San Diego

CPU Undervolting

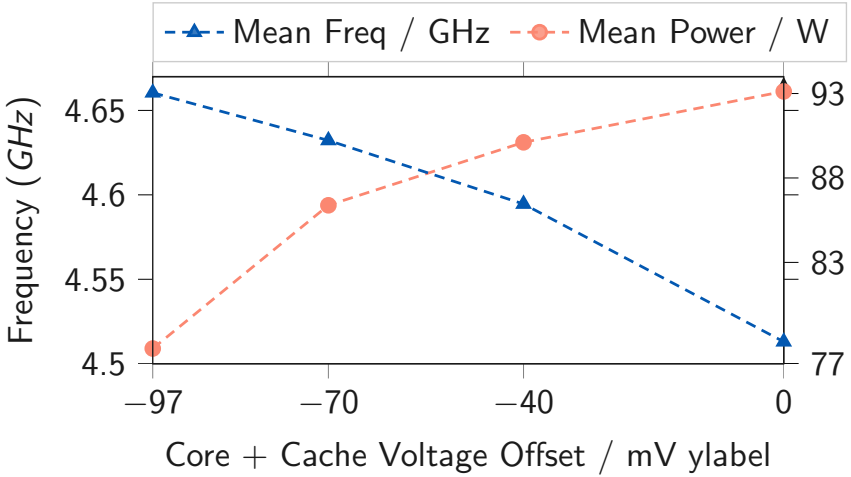
Decrease Power Consumption

Decrease Power Consumption



Increase Performance

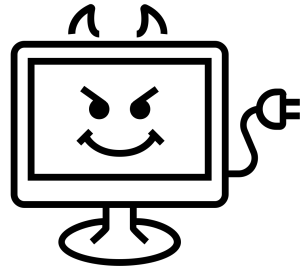
Performance Improvement and Power Savings (as a graph)



Performance Improvement and Power Savings

CPU	V_{off}	Score	Power	Freq.	Energy Eff.
i5-1035G1	−70 mV	+6.0 %	−0.1 %	+8.5 %	+6.1 %
	−97 mV	+7.9 %	−0.5 %	+12 %	+8.4 %
i9-9900K	−70 mV	+2.2 %	−7.2 %	+2.6 %	+10 %
	−97 mV	+3.8 %	−16 %	+3.3 %	+23 %
7700X*	−70 mV	+1.4 %	−9.8 %	+1.8 %	+12 %
	−97 mV	+1.9 %	−15 %	+1.8 %	+20 %

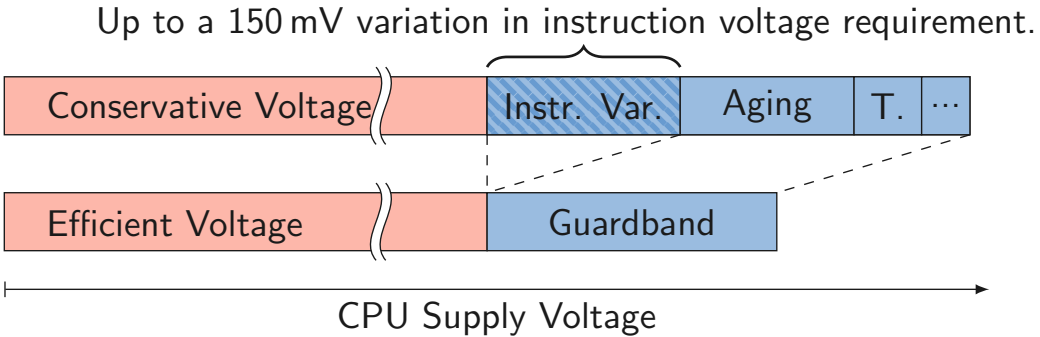
Cause Reliability Issues



Cause **Security** Issues

About Faulting Instructions

CPU Supply Voltage



Some Instruction Produce Faulty Results

Instruction	IMUL	VOR*	AESENC	VXOR*	VANDN*	VAND*	VSQRTPD	VPCLMULQDQ	VPSRAD	VPCMP*	VPMAX*	VPADDQ[1]
Number of Faults	**79**	47	40	40	30	28	24	16	9	5	3	1

[1] A. Kogler, D. Gruss, and M. Schwarz. Minefield: A Software-only Protection for SGX Enclaves against DVFS Attacks. In: USENIX Security. 2022.

CIkscrew by Tang et al. [TSS17]

- ARM TrustZone
- Overclocking

CIkscrew by Tang et al. [TSS17]

- ARM TrustZone
- Overclocking

Voltage Jockey by Qiu et al. [Qiu+19]

- ARM TrustZone
- Undervolting

Clkscrew by Tang et al. [TSS17]

- ARM TrustZone
- Overclocking

Voltage Jockey by Qiu et al. [Qiu+19]

- ARM TrustZone
- Undervolting

Plundervolt by Murdock et al. [Mur+20]

- Intel SGX
- Undervolting

But undervolting could save so much power!

Is it viable to undervolt a CPU while disabling faulting instructions?

Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead?

Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead?
- Are the any significant energy savings?

Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead?
- Are the any significant energy savings?
- Is the CPU still secure?

Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead? ✓
- Are the any significant energy savings?
- Is the CPU still secure?

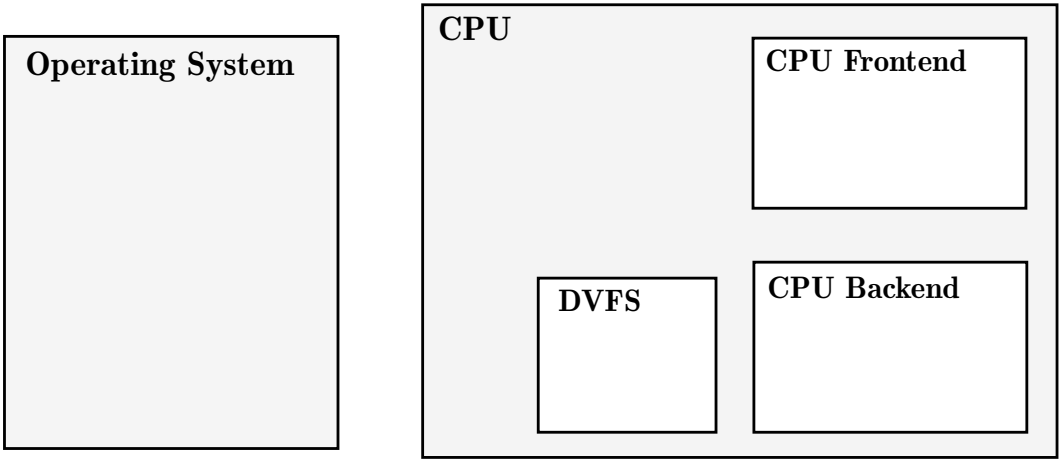
Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead? ✓
- Are the any significant energy savings? ✓
- Is the CPU still secure?

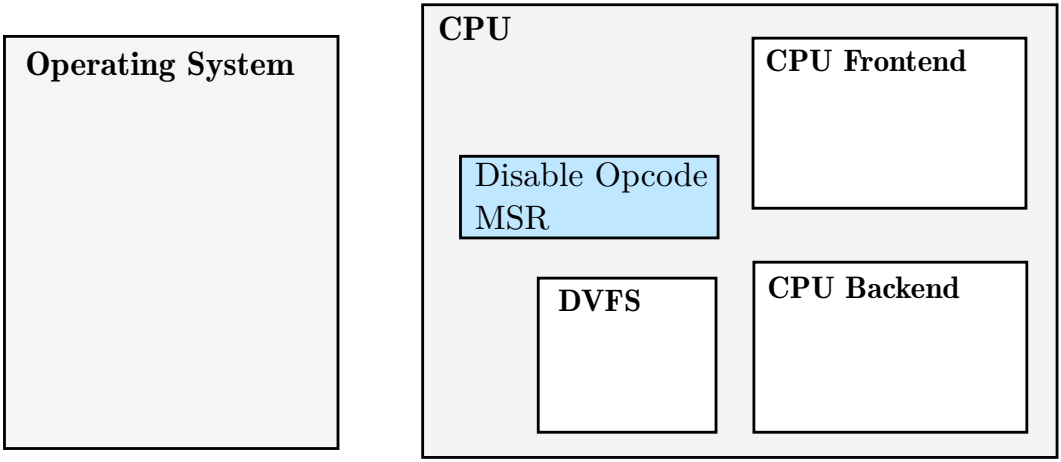
Is it viable to undervolt a CPU while disabling faulting instructions?

- Do the benefits outweigh the overhead? ✓
- Are the any significant energy savings? ✓
- Is the CPU still secure? ✓

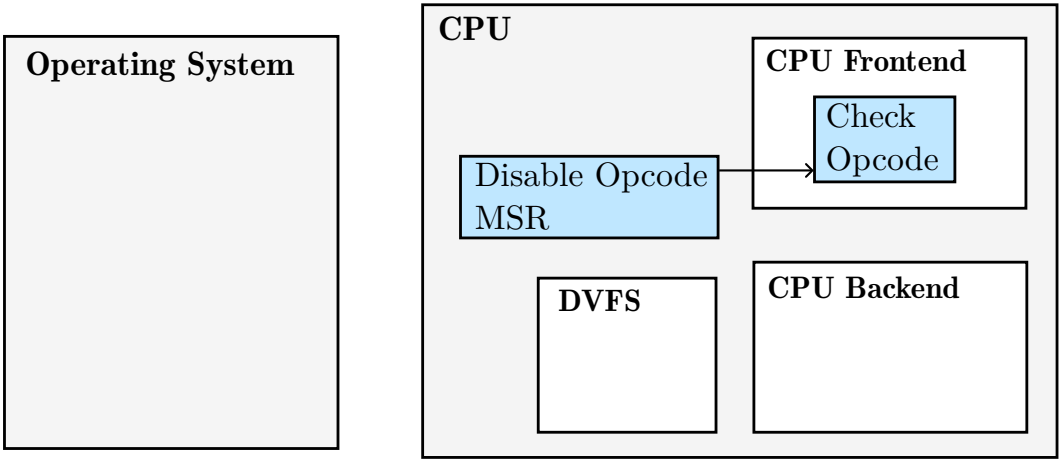
Overview



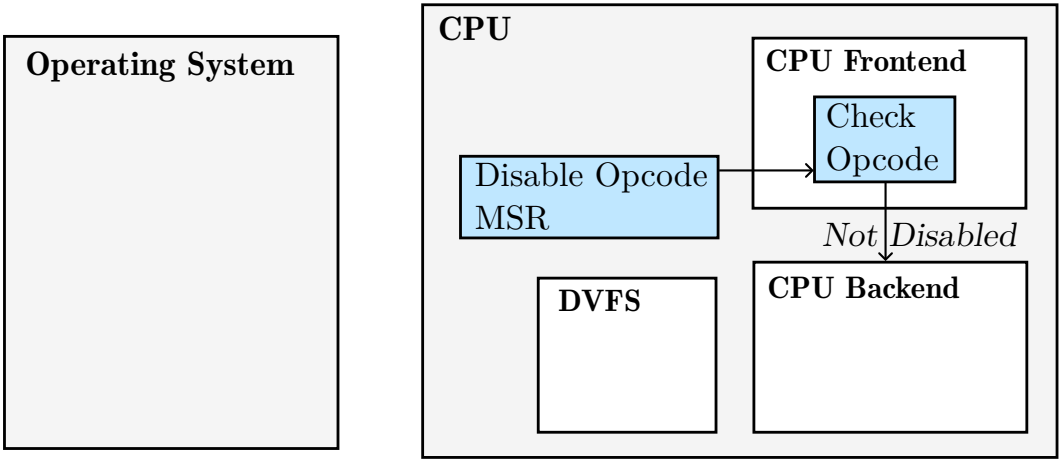
Overview


Operating System
CPU
CPU Frontend
Disable Opcode MSR
DVFS
CPU Backend

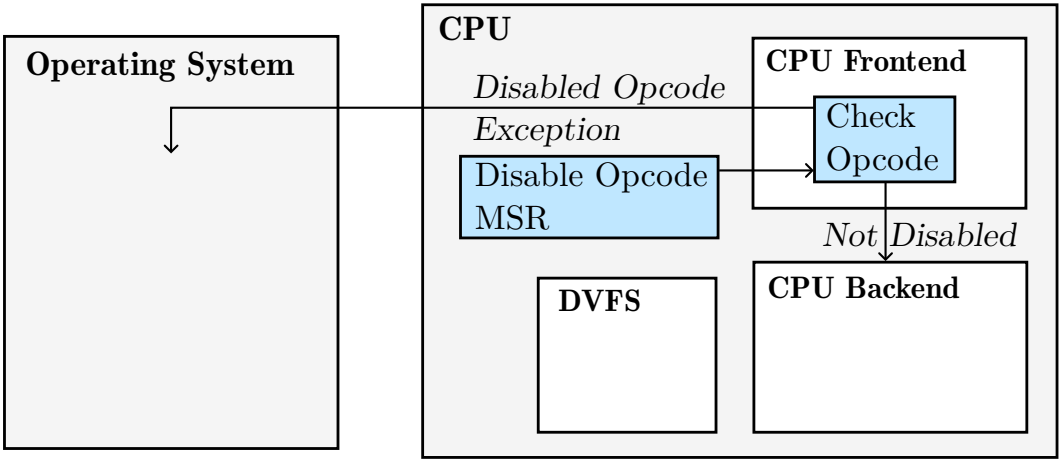
Overview


Operating System
CPU
CPU Frontend
Check Opcode
Disable Opcode MSR
DVFS
CPU Backend

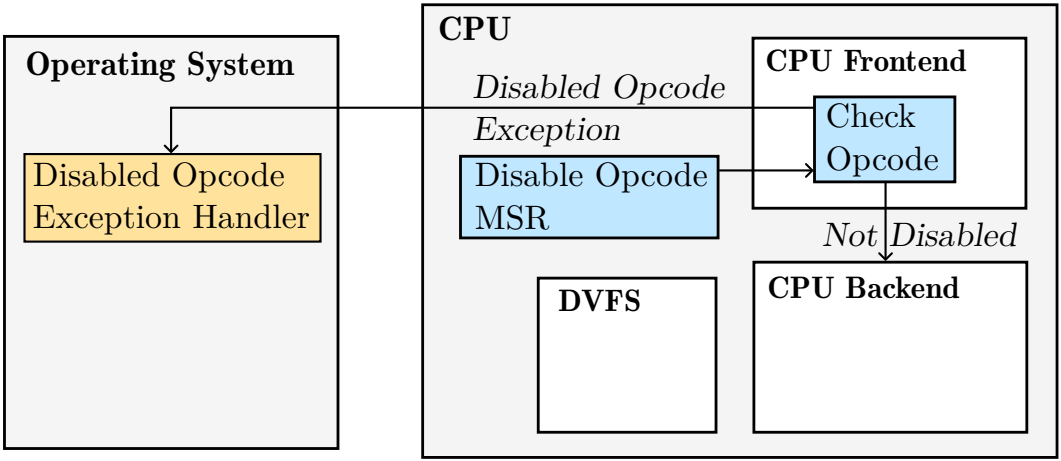
Overview


Operating System
CPU
CPU Frontend
Check Opcode
Disable Opcode MSR
Not Disabled
DVFS
CPU Backend

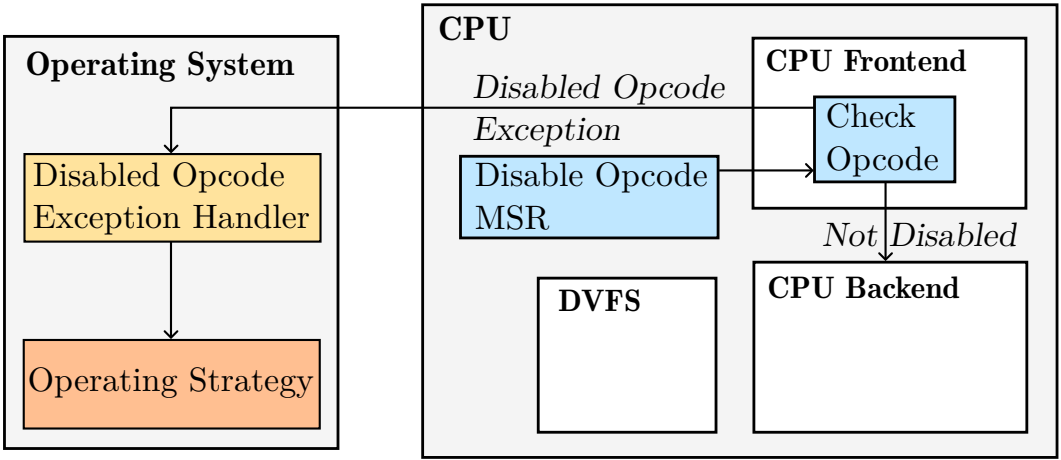
Overview


Operating System
CPU
CPU Frontend
Disabled Opcode
Exception
Check
Opcode
Disable Opcode
MSR
Not Disabled
DVFS
CPU Backend

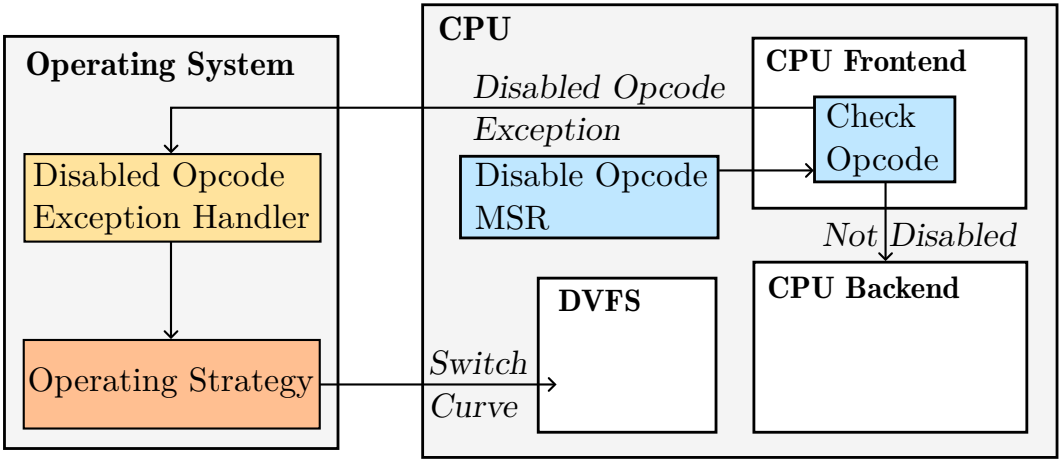
Overview


Operating System
Disabled Opcode Exception Handler
CPU
Disabled Opcode Exception
CPU Frontend
Check Opcode
Disable Opcode MSR
Not Disabled
DVFS
CPU Backend

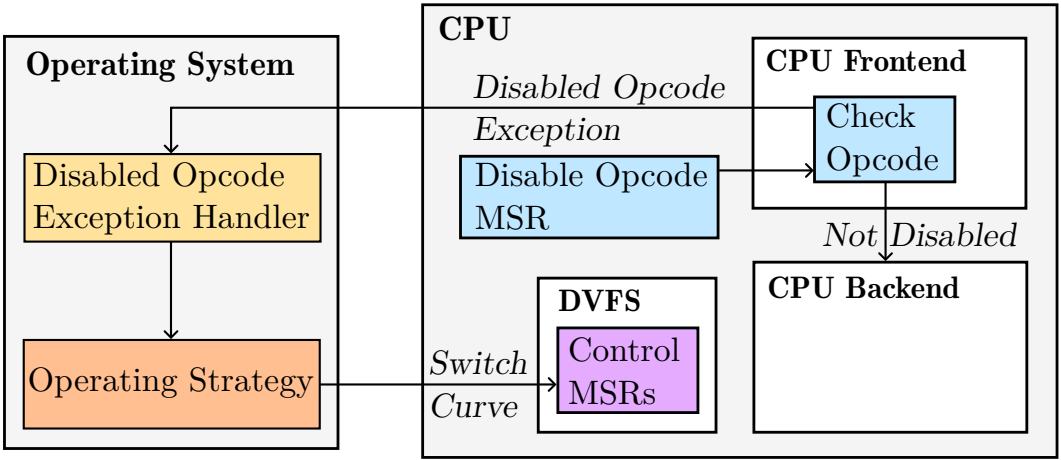
Overview


Operating System
Disabled Opcode Exception Handler
Operating Strategy
CPU
Disabled Opcode Exception
CPU Frontend
Check Opcode
Disable Opcode MSR
Not Disabled
DVFS
CPU Backend

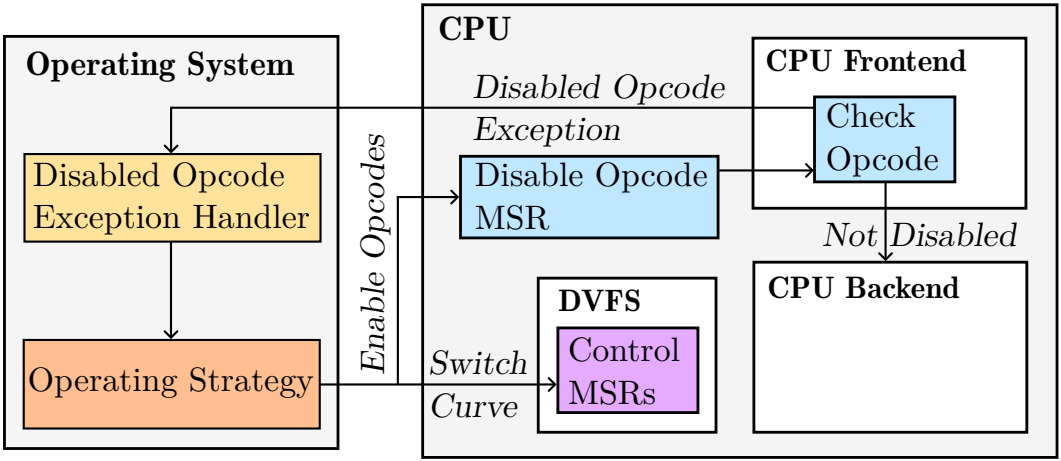
Overview


Operating System
Disabled Opcode Exception Handler
Operating Strategy
CPU
Disabled Opcode Exception
Disable Opcode MSR
CPU Frontend
Check Opcode
Not Disabled
CPU Backend
DVFS
Switch Curve

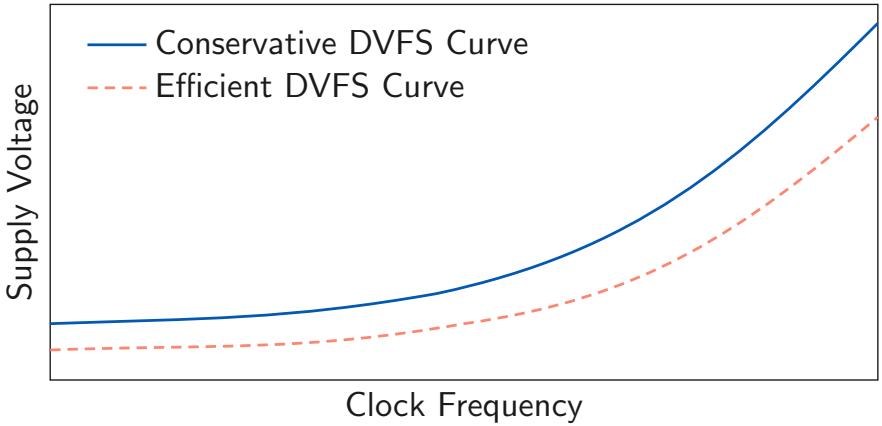
Overview


Operating System
Disabled Opcode Exception Handler
Operating Strategy
CPU
Disabled Opcode Exception
Disable Opcode MSR
CPU Frontend
Check Opcode
Not Disabled
CPU Backend
DVFS
Control MSRs
Switch Curve

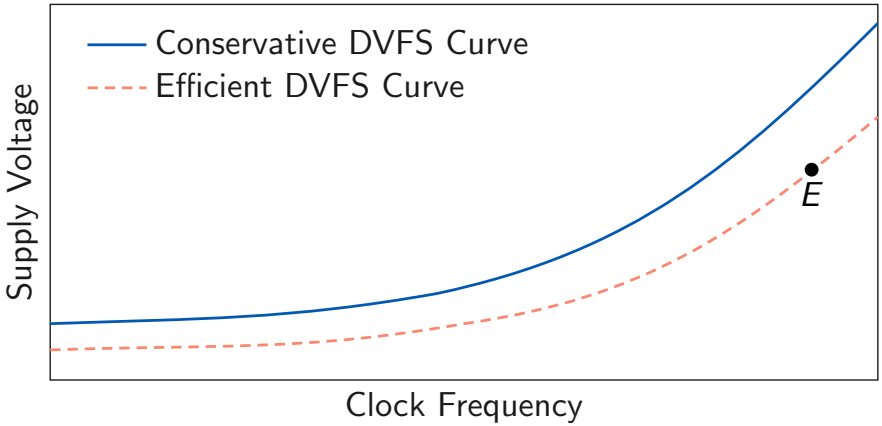
Overview


Operating System
Disabled Opcode Exception Handler
Operating Strategy
Enable Opcodes
CPU
Disabled Opcode Exception
Disable Opcode MSR
Switch Curve
DVFS
Control MSRs
CPU Frontend
Check Opcode
Not Disabled
CPU Backend

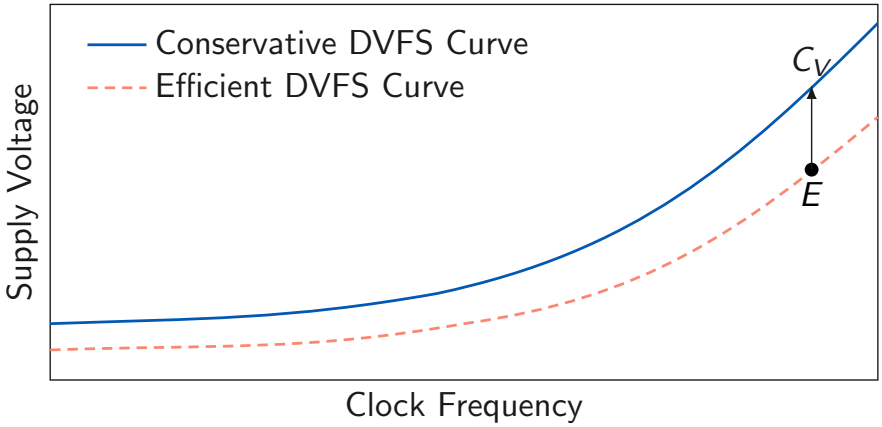
DVFS Curves and Operating Strategies



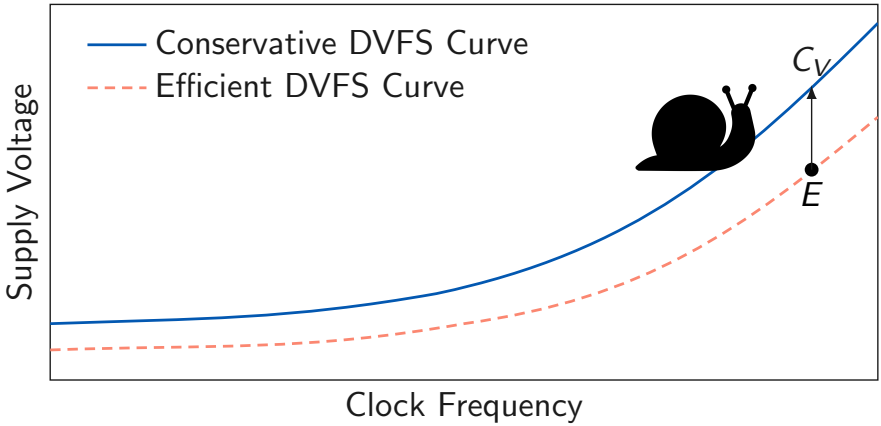
DVFS Curves and Operating Strategies



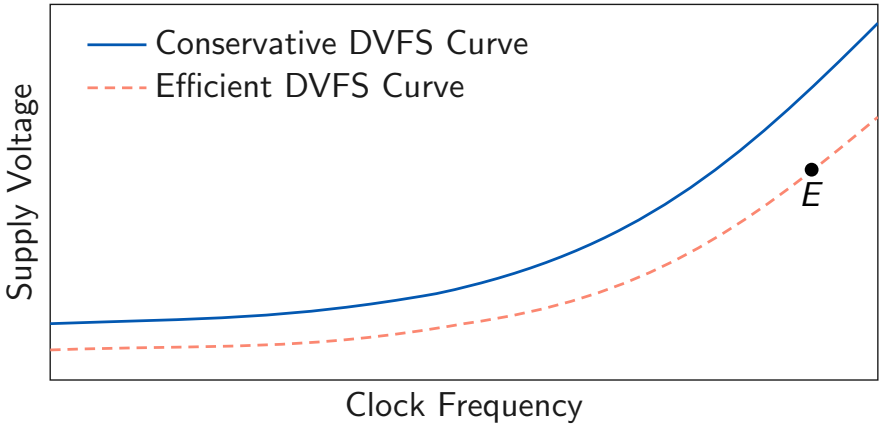
DVFS Curves and Operating Strategies



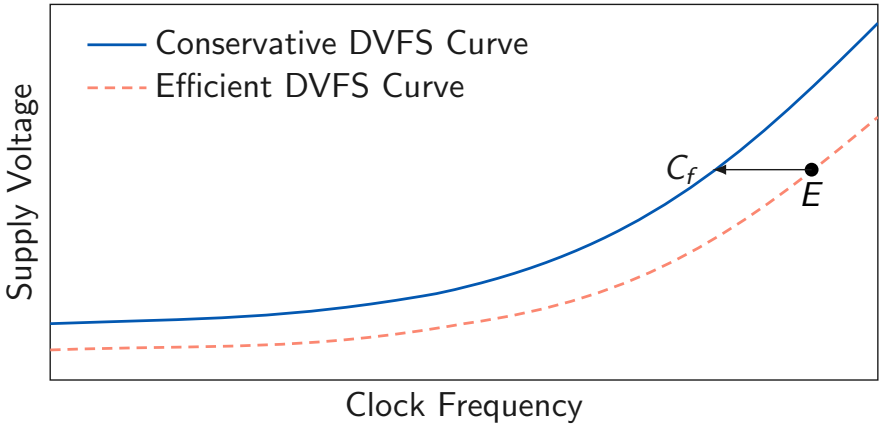
DVFS Curves and Operating Strategies



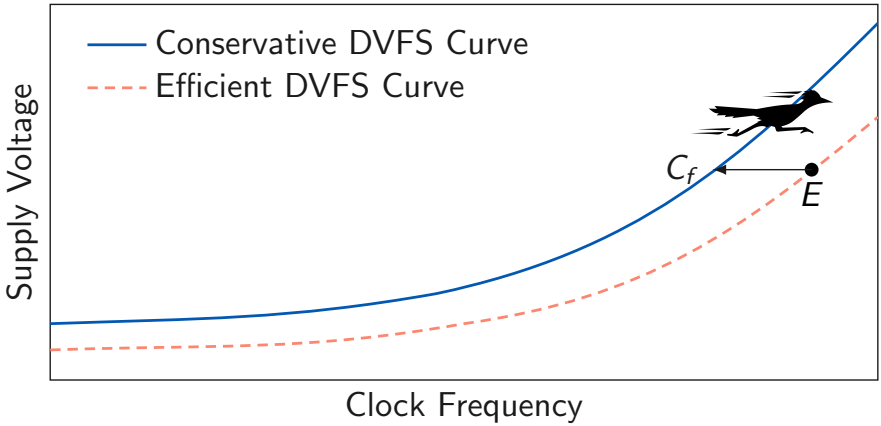
DVFS Curves and Operating Strategies



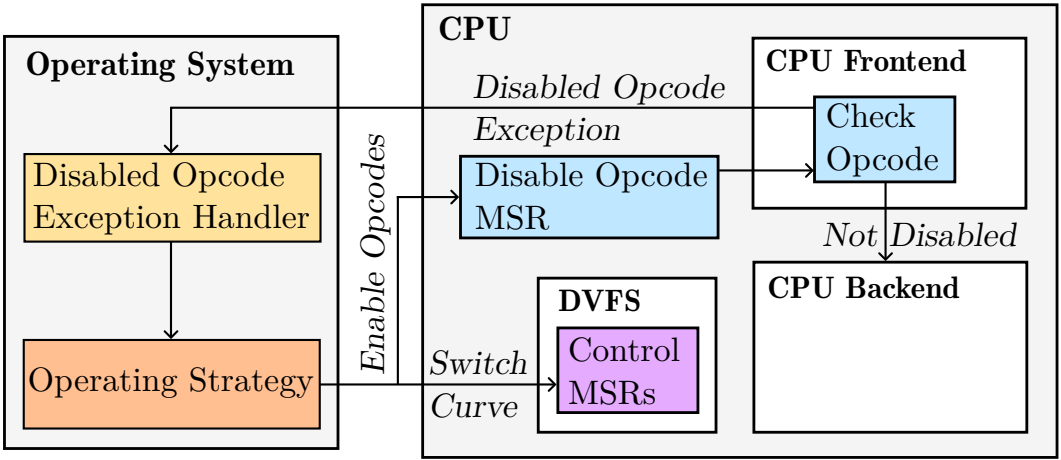
DVFS Curves and Operating Strategies



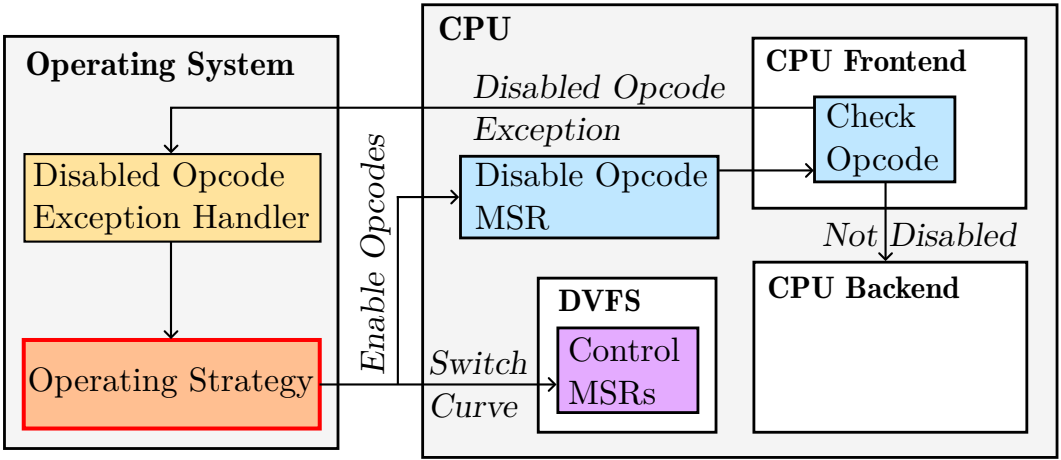
DVFS Curves and Operating Strategies



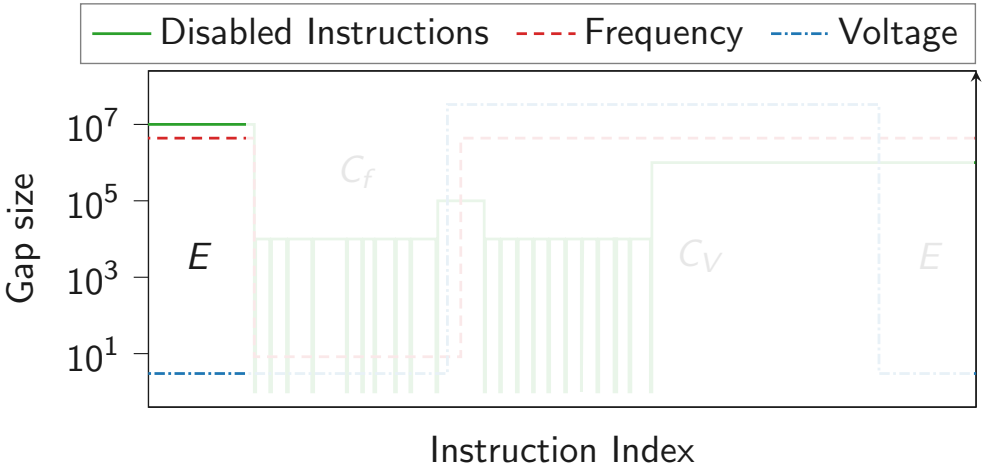
Operating Strategy



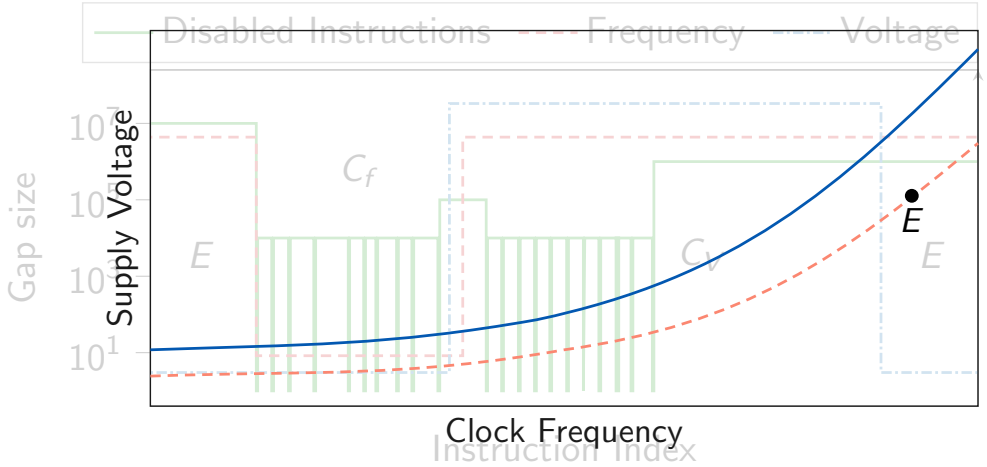
Operating Strategy



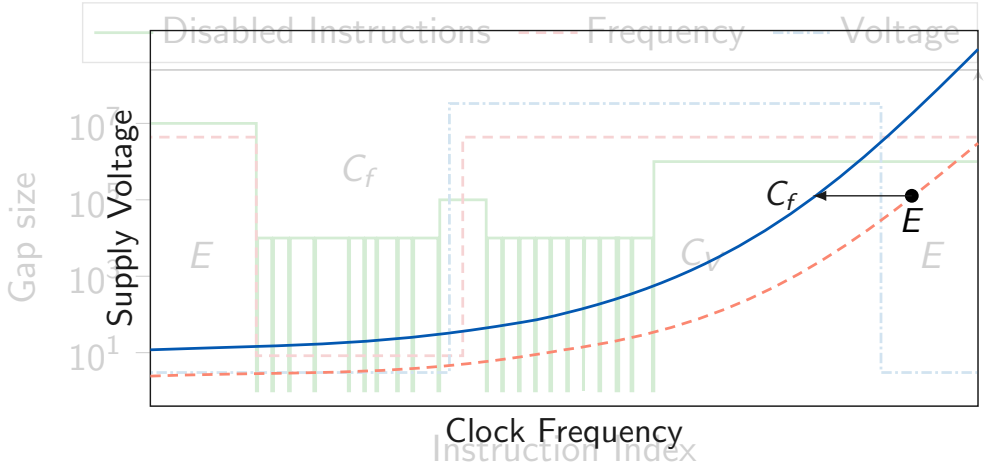
fV Operating Strategy

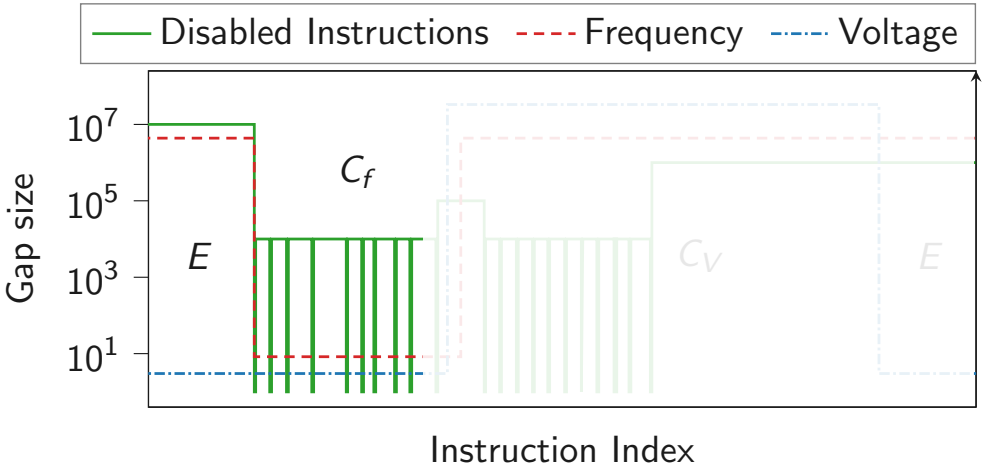


fV Operating Strategy

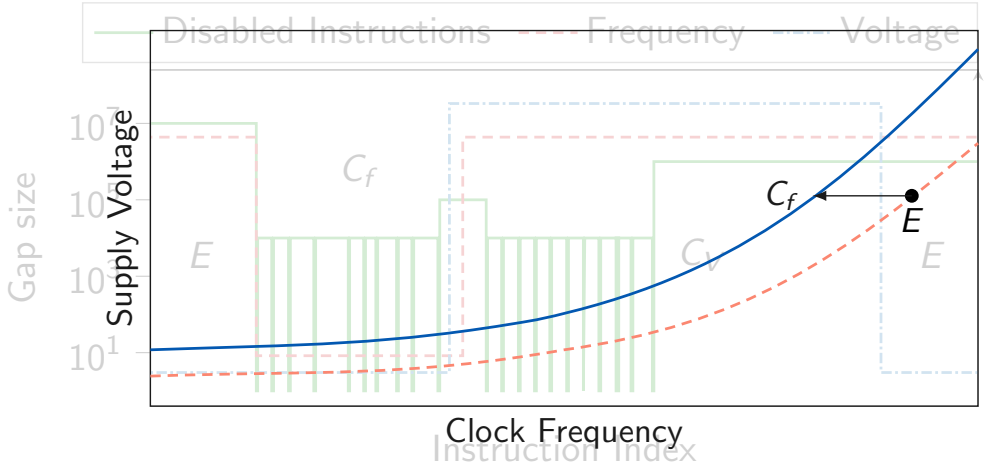


fV Operating Strategy

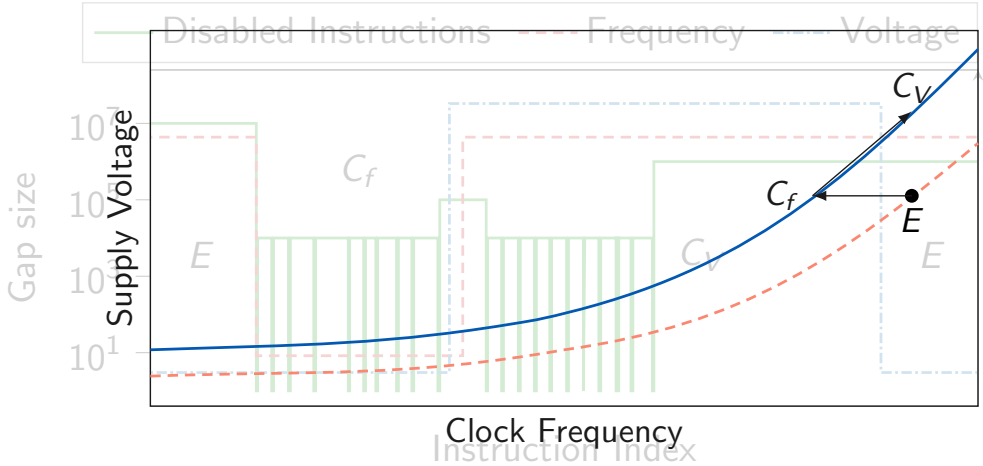



Disabled Instructions
Frequency
Voltage
Gap size
10^7
10^5
10^3
10^1
C_f
E
C_V
E
Instruction Index

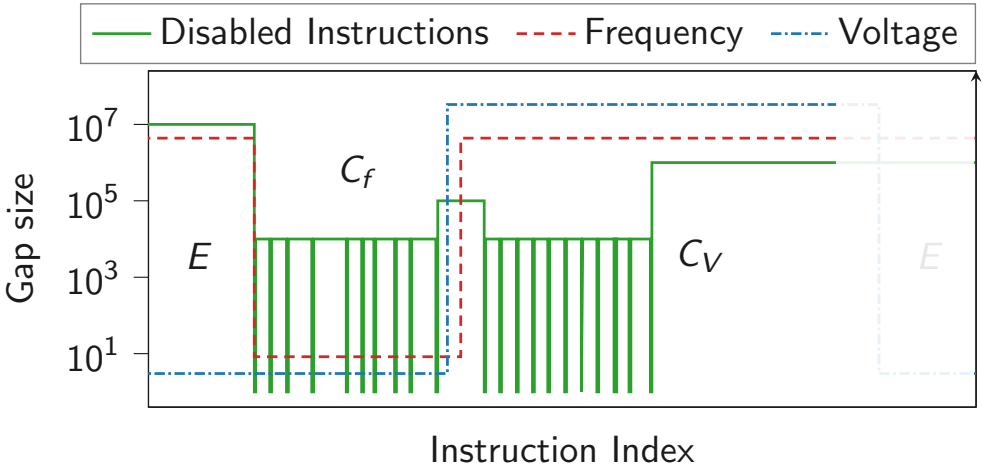
fV Operating Strategy



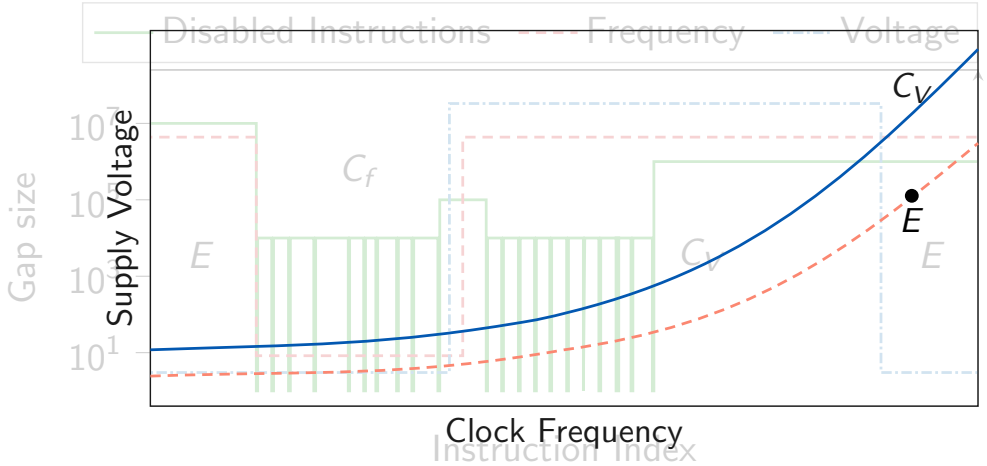
fV Operating Strategy



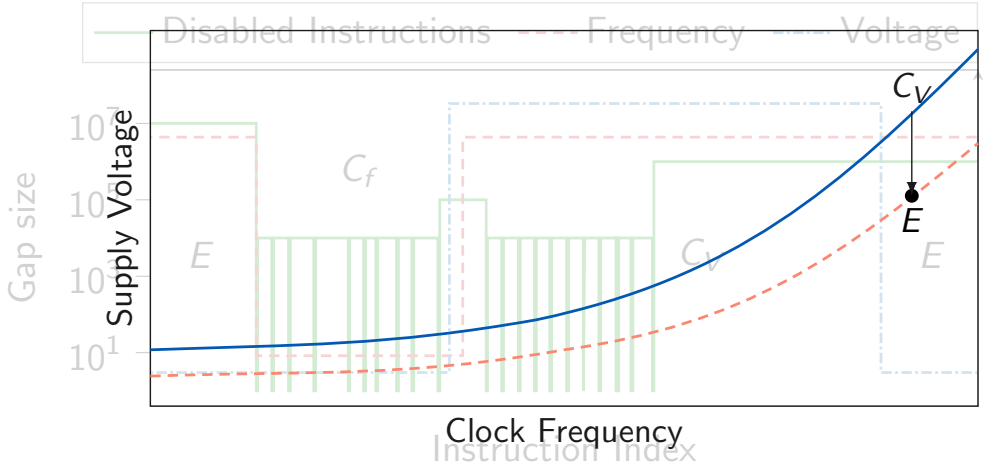
fV Operating Strategy



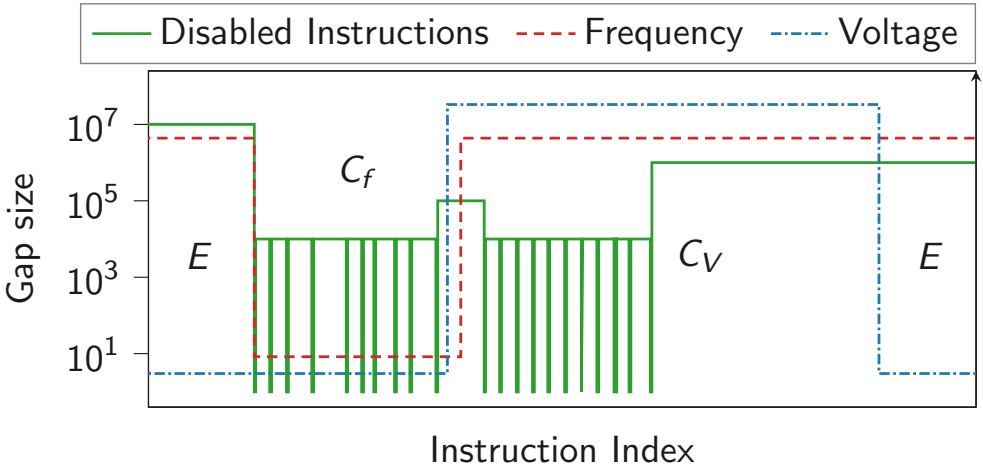
fV Operating Strategy



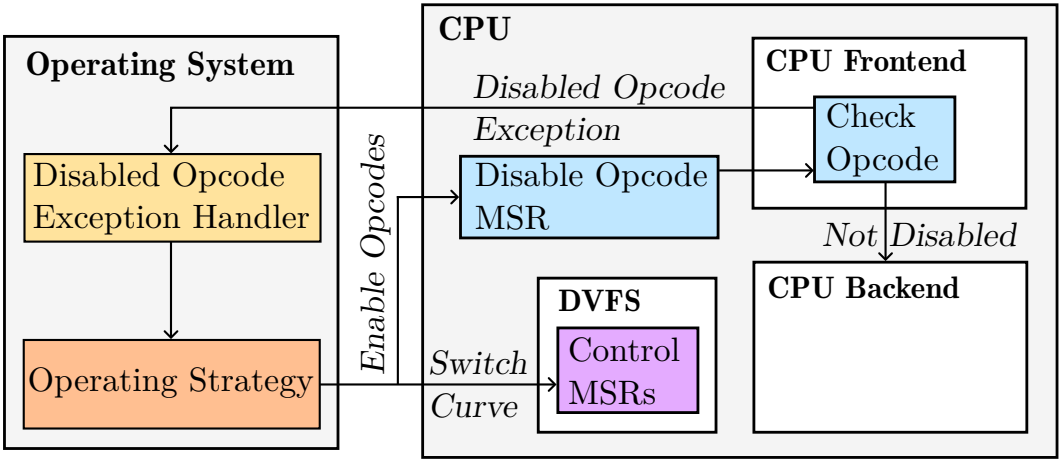
fV Operating Strategy



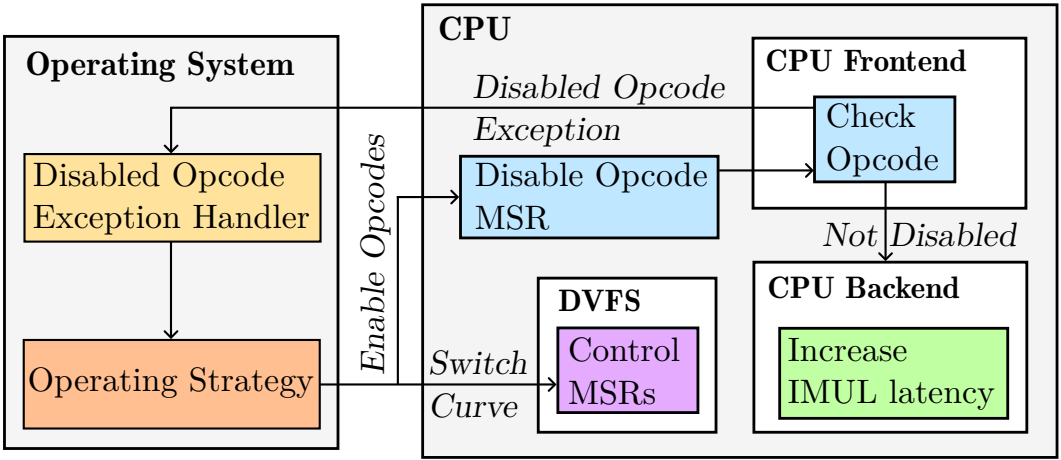
fV Operating Strategy



Increase IMUL Latency

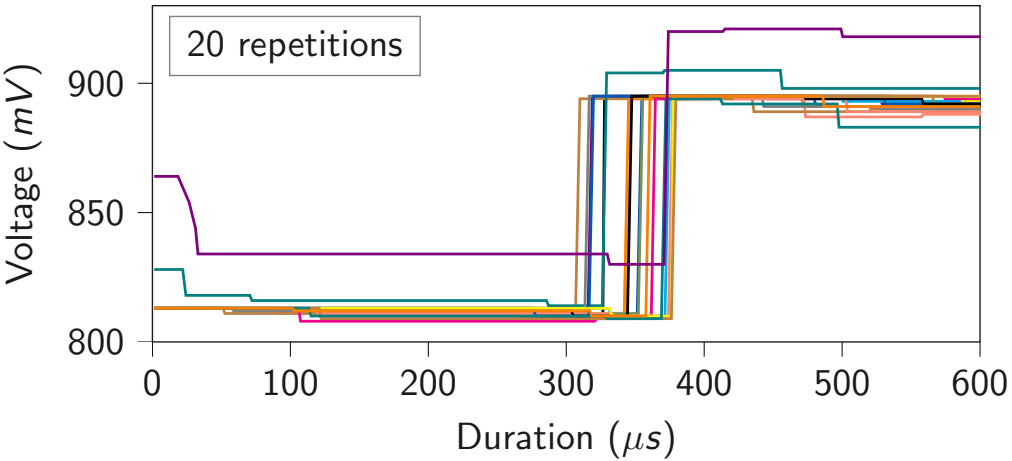


Increase IMUL Latency

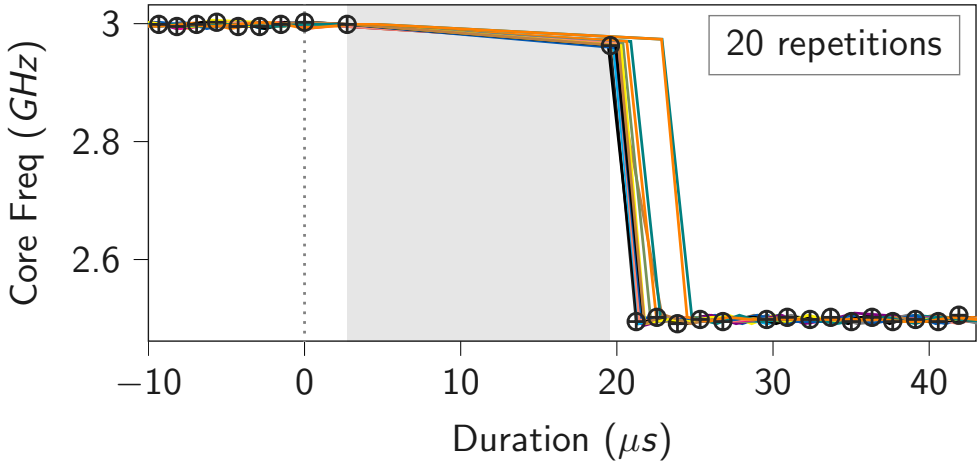


Evaluation

Voltage Change Delay



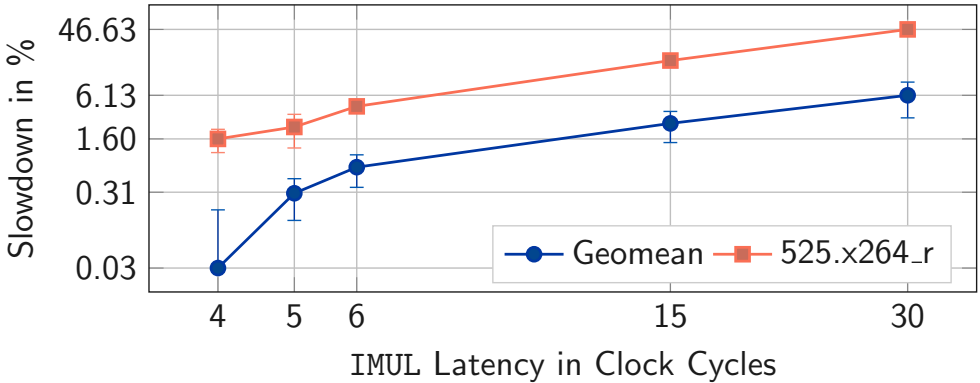
Frequency Change Delay



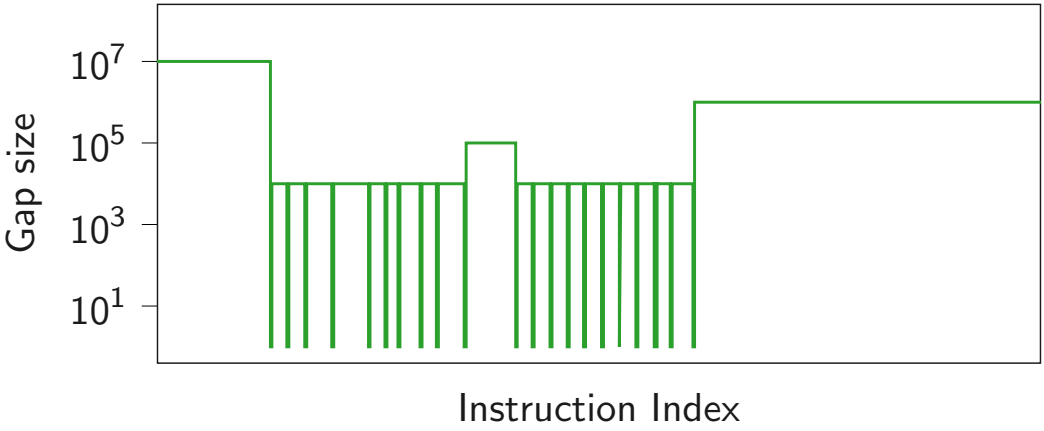
Performance Improvement and Power Savings

CPU	V_{off}	Score	Power	Freq.	Energy Eff.
i5-1035G1	−70 mV	+6.0 %	−0.1 %	+8.5 %	+6.1 %
	−97 mV	+7.9 %	−0.5 %	+12 %	+8.4 %
i9-9900K	−70 mV	+2.2 %	−7.2 %	+2.6 %	+10 %
	−97 mV	+3.8 %	−16 %	+3.3 %	+23 %
7700X*	−70 mV	+1.4 %	−9.8 %	+1.8 %	+12 %
	−97 mV	+1.9 %	−15 %	+1.8 %	+20 %

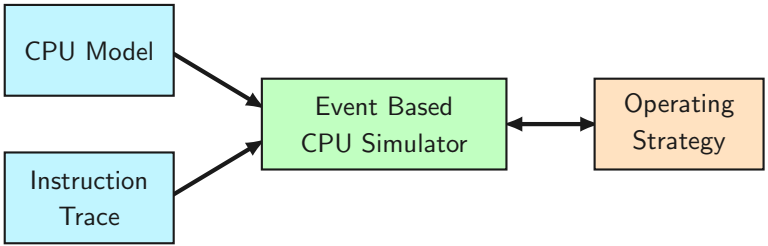
Increasing the IMUL Latency



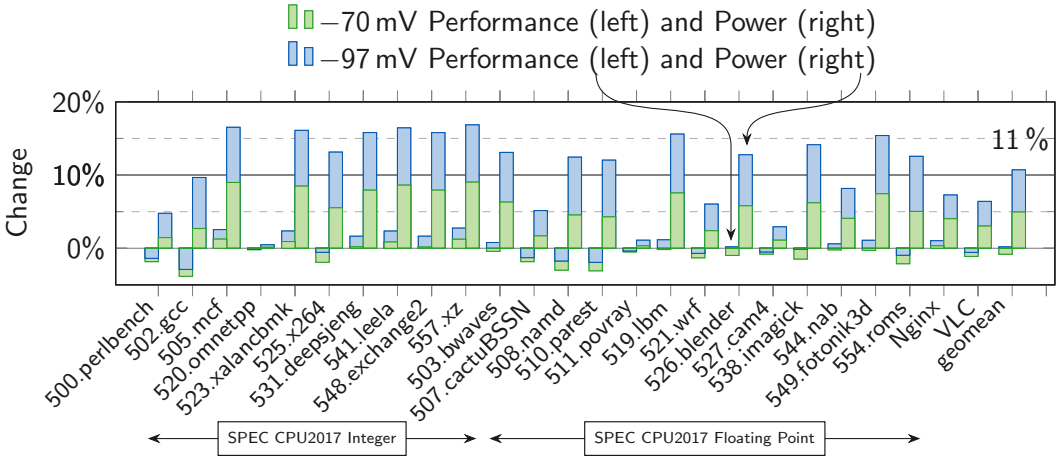
Instruction Traces



Putting Everything Together



SPEC CPU2017 Results



More Results

CPU cores	OS		70 mV Undervolt						97 mV Undervolt					
			SPEC$_{gmean}$	SPEC$_{median}$	525.x264	SPEC$_{noSIMD}$	Nginx	VLC	SPEC$_{gmean}$	SPEC$_{median}$	525.x264	SPEC$_{noSIMD}$	Nginx	VLC
$\mathcal{A}_1$	*fV*	Pwr	−5.6 %	−7.1 %	−7.1 %	−7.1 %	−3.5 %	−3.9 %	−9.7 %	−11 %	−12 %	−15 %	−5.8 %	−6.3 %
		Perf.	−0.2 %	−1.3 %	−1.3 %	+3.0 %	+0.5 %	−0.4 %	+0.8 %	+1.3 %	0.1 %	+3.4 %	+1.2 %	+0.2 %
		Eff.	+5.7 %	+6.2 %	+6.2 %	+11 %	+4.2 %	+3.6 %	+12 %	+14 %	+14 %	+21 %	+7.4 %	+6.9 %
$\mathcal{A}_4$	*fV*	Pwr	−4.6 %	−0.1 %	−6.9 %	−7.4 %	−1.0 %	−1.0 %	−8.9 %	−8.7 %	−13 %	−16 %	−1.6 %	−1.6 %
		Perf.	−3.9 %	−0.0 %	−7.9 %	+1.8 %	−0.3 %	−0.6 %	−3.6 %	−3.5 %	−7.2 %	+1.8 %	−0.1 %	−0.5 %
		Eff.	+0.7 %	0.1 %	−1.0 %	+10.0 %	+0.7 %	+0.4 %	+5.8 %	+5.7 %	+6.7 %	+22 %	+1.5 %	+1.1 %
$\mathcal{A}_\infty$	*e*	Pwr	−7.5 %	−7.6 %	−5.4 %	−7.5 %	−7.2 %	−7.2 %	−12 %	−12 %	−10 %	−17 %	−12 %	−12 %
		Perf.	−42 %	−12 %	+6.2 %	+1.4 %	−98 %	−92 %	−42 %	−12 %	+6.1 %	+1.4 %	−98 %	−92 %
		Eff.	−37 %	−4.5 %	+12 %	+9.6 %	−98 %	−91 %	−34 %	+0.6 %	+18 %	+22 %	−98 %	−91 %
$\mathcal{B}_\infty$	*f*	Pwr	−8.1 %	−7.8 %	−7.8 %	−9.1 %	−4.4 %	−4.4 %	−12 %	−11 %	−11 %	−14 %	−6.7 %	−6.7 %
		Perf.	−7.8 %	−7.8 %	−9.2 %	+0.4 %	−2.5 %	−2.5 %	−10 %	−11 %	−12 %	+0.6 %	−2.3 %	−2.3 %
		Eff.	+0.3 %	−0.0 %	−1.6 %	+11 %	+2.0 %	+2.0 %	+1.4 %	0.1 %	−1.6 %	+17 %	+4.7 %	+4.7 %
	e	Pwr	−9.2 %	−8.0 %	−11 %	−9.2 %	−9.8 %	−9.8 %	−14 %	−13 %	−16 %	−14 %	−15 %	−15 %
		Perf.	−26 %	−5.1 %	+15 %	−0.5 %	−96 %	−80 %	−26 %	−5.2 %	+19 %	0.0 %	−96 %	−80 %
		Eff.	−19 %	+3.1 %	+28 %	+9.5 %	−95 %	−78 %	−14 %	+9.3 %	+41 %	+17 %	−95 %	−76 %
$\mathcal{C}_\infty$	*fV*	Pwr	−5.6 %	−7.1 %	−7.1 %	−6.1 %	−3.6 %	−4.0 %	−9.8 %	−11 %	−12 %	−14 %	−5.8 %	−6.6 %
		Perf.	−0.8 %	−1.9 %	−1.9 %	+3.5 %	+0.3 %	−1.1 %	+0.2 %	+0.2 %	−0.6 %	+3.8 %	+1.0 %	−0.6 %
		Eff.	+5.1 %	+5.5 %	+5.5 %	+10 %	+4.0 %	+3.0 %	+11 %	+13 %	+13 %	+21 %	+7.3 %	+6.4 %

5.2 %	+11 %	+4.2 %	+3.6 %	+12 %	+14 %	+14 %	+21 %	+7.4 %	+6.9 %
6.9 %	−7.4 %	−1.0 %	−1.0 %	−8.9 %	−8.7 %	−13 %	−16 %	−1.6 %	−1.6 %
7.9 %	+1.8 %	−0.3 %	−0.6 %	−3.6 %	−3.5 %	−7.2 %	+1.8 %	−0.1 %	−0.5 %
1.0 %	+10.0 %	+0.7 %	+0.4 %	+5.8 %	+5.7 %	+6.7 %	+22 %	+1.5 %	+1.1 %
5.4 %	−7.5 %	−7.2 %	−7.2 %	−12 %	−12 %	−10 %	−17 %	−12 %	−12 %
6.2 %	+1.4 %	−98 %	−92 %	−42 %	−12 %	+6.1 %	+1.4 %	−98 %	−92 %
12 %	+9.6 %	−98 %	−91 %	−34 %	+0.6 %	+18 %	+22 %	−98 %	−91 %
7.8 %	−9.1 %	−4.4 %	−4.4 %	−12 %	−11 %	−11 %	−14 %	−6.7 %	−6.7 %
9.2 %	+0.4 %	−2.5 %	−2.5 %	−10 %	−11 %	−12 %	+0.6 %	−2.3 %	−2.3 %
1.6 %	+11 %	+2.0 %	+2.0 %	+1.4 %	0.1 %	−1.6 %	+17 %	+4.7 %	+4.7 %
11 %	−9.2 %	−9.8 %	−9.8 %	−14 %	−13 %	−16 %	−14 %	−15 %	−15 %
15 %	−0.5 %	−96 %	−80 %	−26 %	−5.2 %	+19 %	0.0 %	−96 %	−80 %
28 %	+9.5 %	−95 %	−78 %	−14 %	+9.3 %	+41 %	+17 %	−95 %	−76 %
7.1 %	−6.1 %	−3.6 %	−4.0 %	−9.8 %	−11 %	−12 %	−14 %	−5.8 %	−6.6 %
1.9 %	+3.5 %	+0.3 %	−1.1 %	+0.2 %	+0.2 %	−0.6 %	+3.8 %	+1.0 %	−0.6 %
5.5 %	+10 %	+4.0 %	+3.0 %	+11 %	+13 %	+13 %	+21 %	+7.3 %	+6.4 %

5.2 %	+11 %	+4.2 %	+3.6 %	+12 %	+14 %	+14 %	+21 %	+7.4 %	+6.9 %
6.9 %	−7.4 %	−1.0 %	−1.0 %	−8.9 %	−8.7 %	−13 %	−16 %	−1.6 %	−1.6 %
7.9 %	+1.8 %	−0.3 %	−0.6 %	−3.6 %	−3.5 %	−7.2 %	+1.8 %	−0.1 %	−0.5 %
1.0 %	+10.0 %	+0.7 %	+0.4 %	+5.8 %	+5.7 %	+6.7 %	+22 %	+1.5 %	+1.1 %
5.4 %	−7.5 %	−7.2 %	−7.2 %	−12 %	−12 %	−10 %	−17 %	−12 %	−12 %
6.2 %	+1.4 %	−98 %	−92 %	−42 %	−12 %	+6.1 %	+1.4 %	−98 %	−92 %
12 %	+9.6 %	−98 %	−91 %	−34 %	+0.6 %	+18 %	+22 %	−98 %	−91 %
7.8 %	−9.1 %	−4.4 %	−4.4 %	−12 %	−11 %	−11 %	−14 %	−6.7 %	−6.7 %
9.2 %	+0.4 %	−2.5 %	−2.5 %	−10 %	−11 %	−12 %	+0.6 %	−2.3 %	−2.3 %
1.6 %	+11 %	+2.0 %	+2.0 %	+1.4 %	0.1 %	−1.6 %	+17 %	+4.7 %	+4.7 %
11 %	−9.2 %	−9.8 %	−9.8 %	−14 %	−13 %	−16 %	−14 %	−15 %	−15 %
15 %	−0.5 %	−96 %	−80 %	−26 %	−5.2 %	+19 %	0.0 %	−96 %	−80 %
28 %	+9.5 %	−95 %	−78 %	−14 %	+9.3 %	+41 %	+17 %	−95 %	−76 %
7.1 %	−6.1 %	−3.6 %	−4.0 %	−9.8 %	−11 %	−12 %	−14 %	−5.8 %	−6.6 %
1.9 %	+3.5 %	+0.3 %	−1.1 %	+0.2 %	+0.2 %	−0.6 %	+3.8 %	+1.0 %	−0.6 %
5.5 %	+10 %	+4.0 %	+3.0 %	+11 %	+13 %	+13 %	+21 %	+7.3 %	+6.4 %

Final Remarks

"Undervolting" can made secure

Final Remarks

"Undervolting" can made secure

If you want to learn more about:

Final Remarks

"Undervolting" can made secure

If you want to learn more about:

- Analysis of aging and temperature guardband
- Exception delay measurement
- Emulating faulting instruction
- Compiling without faulting instruction
- Detailed security analysis

Final Remarks

"Undervolting" can made secure

If you want to learn more about:

- Analysis of aging and temperature guardband
- Exception delay measurement
- Emulating faulting instruction
- Compiling without faulting instr[illegible]on
- Detailed security analysis

Read the Paper

SUIT

Secure Undervolting with Instruction Traps

Jonas Juffinger, Stepan Kalinin, Daniel Gruss, Frank Mueller

ASPLOS 2024, San Diego, USA — April 27- May 1, 2024

jonas.juffinger@iaik.tugraz.at @notimaginary_ www.jonasjuffinger.com

References i

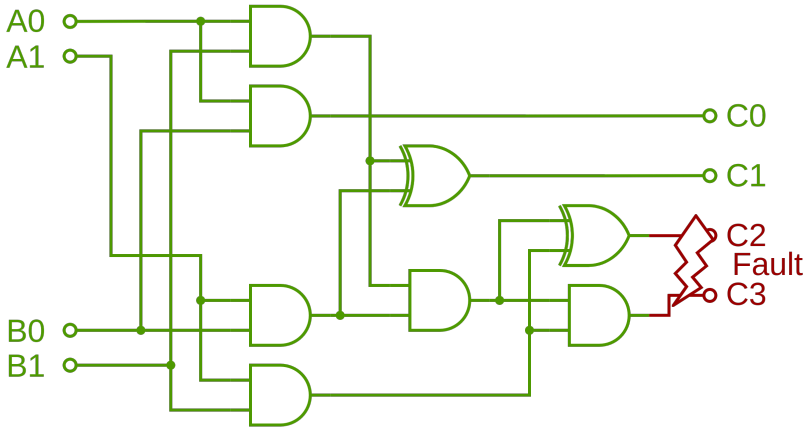
[KGS22] A. Kogler, D. Gruss, and M. Schwarz. Minefield: A Software-only Protection for SGX Enclaves against DVFS Attacks. In: USENIX Security. 2022.

[Mur+20] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens. Plundervolt: Software-based Fault Injection Attacks against Intel SGX. In: S&P. 2020.

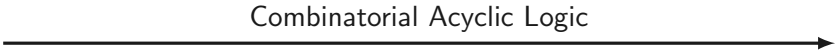
[Qiu+19] P. Qiu, D. Wang, Y. Lyu, and G. Qu. VoltJockey: Breaking SGX by Software-Controlled Voltage-Induced Hardware Faults. In: AsianHOST. 2019.

[TSS17] A. Tang, S. Sethumadhavan, and S. Stolfo. CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management. In: USENIX Security. 2017.

Increase IMUL Latency



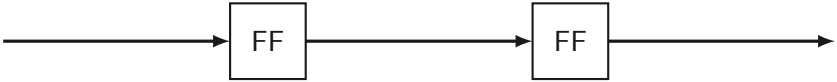
Increase IMUL Latency



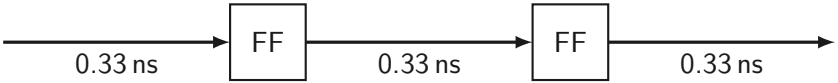
1 GHz

Increase IMUL Latency

Increase IMUL Latency

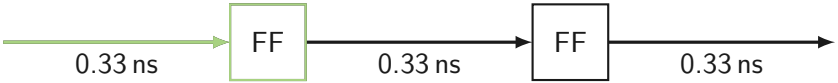


Increase IMUL Latency

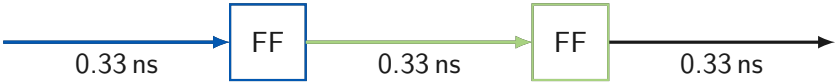


3 GHz

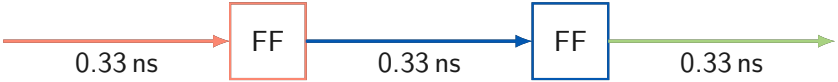
Increase IMUL Latency



Increase IMUL Latency



Increase IMUL Latency



Latency: 3, Throughput: 1

An infinite loop...

```
uint64_t multiplier = 0x1276af93a60d1cb7;
uint64_t var = 0x1313f7d45dd0339a * multiplier;

while (var == 0x1313f7d45dd0339a * multiplier)
{
  var  = 0x1313f7d45dd0339a;
  var *= multiplier;
}

printf("loop exited!\n");
```