VoltJockey: Software-Controlled Voltage-Induced Hardware Fault Injection

Gang Qu
University of Maryland, College Park

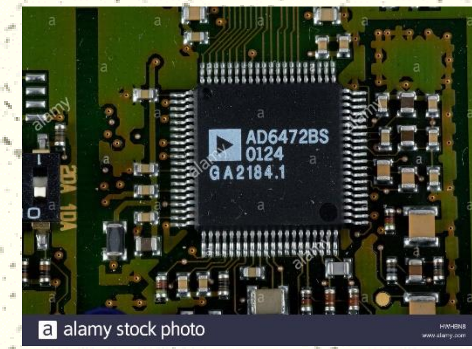
Electronic Design Process Symposium
October 6, 2022
Milpitas, CA

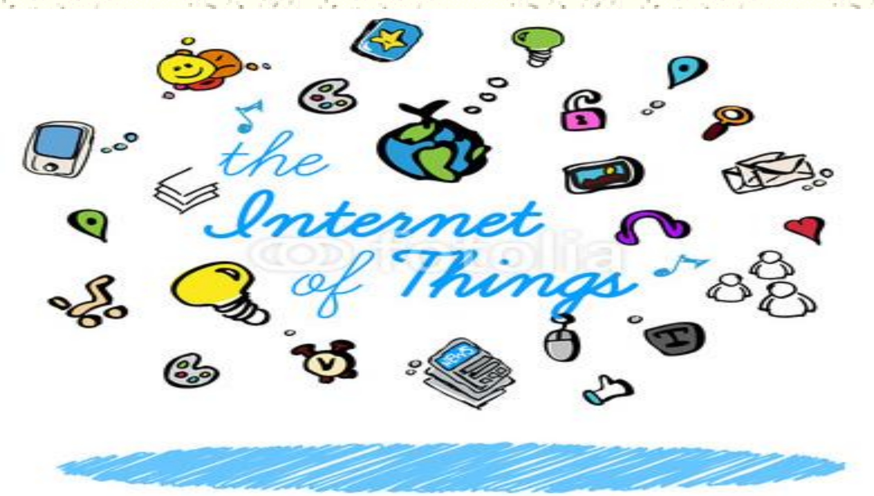
MeshSec Lab

VoltJockey

Why Low Power?

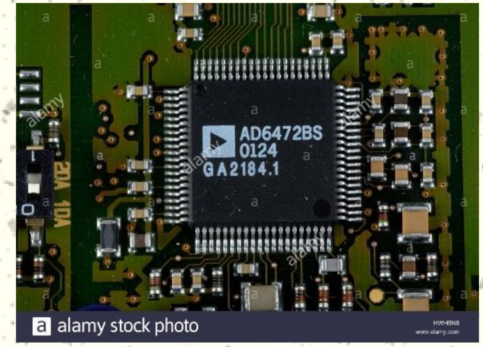
- Longer battery lifetime
- Less packaging/cooling cost
- More reliable circuitry
- Smaller electricity bill





Where Does the Power Go?

- Dynamic power or switching power
- Static power or leakage current
  - Gate-oxide leakage
  - Subthreshold leakage
- Short-circuit power



$$P = \frac{1}{2}\alpha C V_{dd}^2 f + I_{leak} V_{dd} + \alpha Q_{SC} V_{dd} f$$

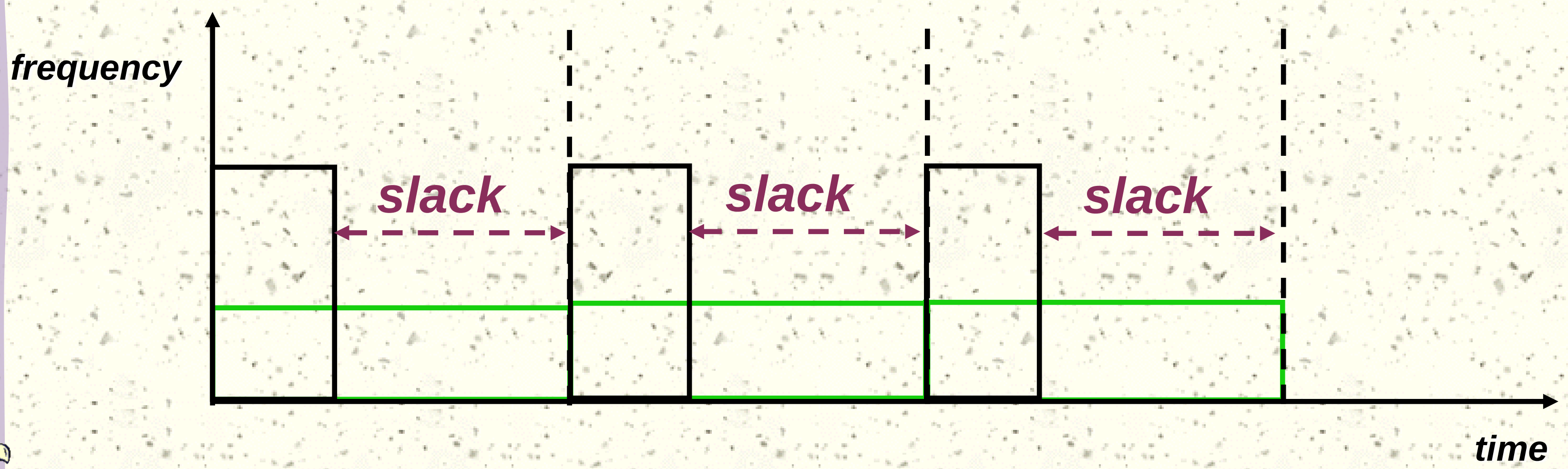
What is DVFS?

- Dynamic voltage and frequency scaling
  - Circuits can work at a range of V_{dd} values
  - A given V_{dd} can support a range of clock frequencies with a
  - $F \propto (V_{dd}-V_{th})^{\chi}/V_{dd}$ $\chi \in (1.0,2.0)$
- Why DVFS saves power and energy?
  - Reduce V_{dd} to γV_{dd}
  - f_{max} reduces to roughly $\gamma\ f_{max}$
  - Dynamic power reduces by roughly γ^3
  - Energy reduces by roughly γ^2

I. Hong, et al. "Power Optimization of Variable Voltage Core-based Systems", DAC'1998.

How Does DVS Work?

- Suppose that a data sample comes every 1 ms
- Requires processing time of 250 μs at 600MHz
- DVS: reduce voltage such that clock slows down to 150MHz



Our Work on Low Power DVFS

1997: variable voltage processor scheduling

1998: M.S. thesis, Variable voltage system (DAC), communication pipeline (ICCAD), real-time scheduling (RTSS).

2000: Quality-energy tradeoff (ISLPED)

2001: limit of energy saving by DVFS (ICCAD)

2002: secure sensor network (ASAP)

2003: multimedia system (ASPDAC, RSP, DAC), probabilistic design (DAC), multi-processor scheduling (EMSOFT), voltage set-up (ICCAD)

2004: performance gain vs energy saving (ISCAS), dual-voltage on (m, k)-firm system (CASES)

2005: parallelism on multi-processor (ASPDAC)

2006: dual-processor fault-tolerant system (ASAP)

2007: leakage aware DVS (AHS), multi-core system scheduling (McSoC)

2013: temperature-aware DVS (ASAP)

Our Work on Low Power DVFS

1997: variable voltage processor scheduling

1998: M.S. thesis, Variable voltage system (DAC), communication pipeline (ICCAD), real-time scheduling (RTSS).

2000: Quality-energy tradeoff (ISLPED)

2001: limit of energy saving by DVFS (ICCAD)

2002: secure sensor network (ASAP)

2003: multimedia system (ASPDAC, RSP, DAC), probabilistic design (DAC), multi-processor scheduling (EMSOFT), voltage set-up (ICCAD)

2004: performance gain vs energy saving (ISCAS), dual-voltage on (m, k)-firm system (CASES)

2005: parallelism on multi-processor (ASPDAC)

2006: dual-processor fault-tolerant system (ASAP)

2007: leakage aware DVS (AHS), multi-core system scheduling (McSoC)

2013: temperature-aware DVS (ASAP)

Our Work on Low Power DVFS

1997: variable voltage processor scheduling
1998: M.S. thesis, Variable voltage system (DAC), communication pipeline (ICCAD), real-time scheduling (RTSS).
2000: Quality-energy tradeoff (ISLPED)
2001: limit of energy saving by DVFS (ICCAD)
2002: secure sensor network (ASAP)
2003: multimedia system (ASPDAC, RSP, DAC), probabilistic design (DAC), multi-processor scheduling (EMSOFT), voltage set-up (ICCAD)
2004: performance gain vs energy saving (ISCAS), dual-voltage on (m, k)-firm system (CASES)
2005: parallelism on multi-processor (ASPDAC)
2006: dual-processor fault-tolerant system (ASAP)
2007: leakage aware DVS (AHS), multi-core system scheduling (McSoC)
2013: temperature-aware DVS (ASAP)

Our Work on Low Power DVFS

1997: variable voltage processor scheduling

1998: M.S. thesis, Variable voltage system (DAC), communication pipeline (ICCAD), real-time scheduling (RTSS).

2000: Quality-energy tradeoff (ISLPED)

2001: limit of energy saving by DVFS (ICCAD)

2002: secure sensor network (ASAP)

2003: multimedia system (ASPDAC, RSP, DAC), probabilistic design (DAC), multi-processor scheduling (EMSOFT), voltage set-up (ICCAD)

2004: performance gain vs energy saving (ISCAS), dual-voltage on (m, k)-firm system (CASES)

2005: parallelism on multi-processor (ASPDAC)

2006: dual-processor fault-tolerant system (ASAP)

2007: leakage aware DVS (AHS), multi-core system scheduling (McSoC)

2013: temperature-aware DVS (ASAP)

Our Work on Low Power DVFS

1997: variable voltage processor scheduling

1998: M.S. thesis, Variable voltage system (DAC), communication pipeline (ICCAD), real-time scheduling (RTSS).

2000: Quality-energy tradeoff (ISLPED)

2001: limit of energy saving by DVFS (ICCAD)

2002: secure sensor network (ASAP)

2003: multimedia system (ASPDAC, RSP, DAC), probabilistic design (DAC), multi-processor scheduling (EMSOFT), voltage set-up (ICCAD)

2004: performance gain vs energy saving (ISCAS), dual-voltage on (m, k)-firm system (CASES)

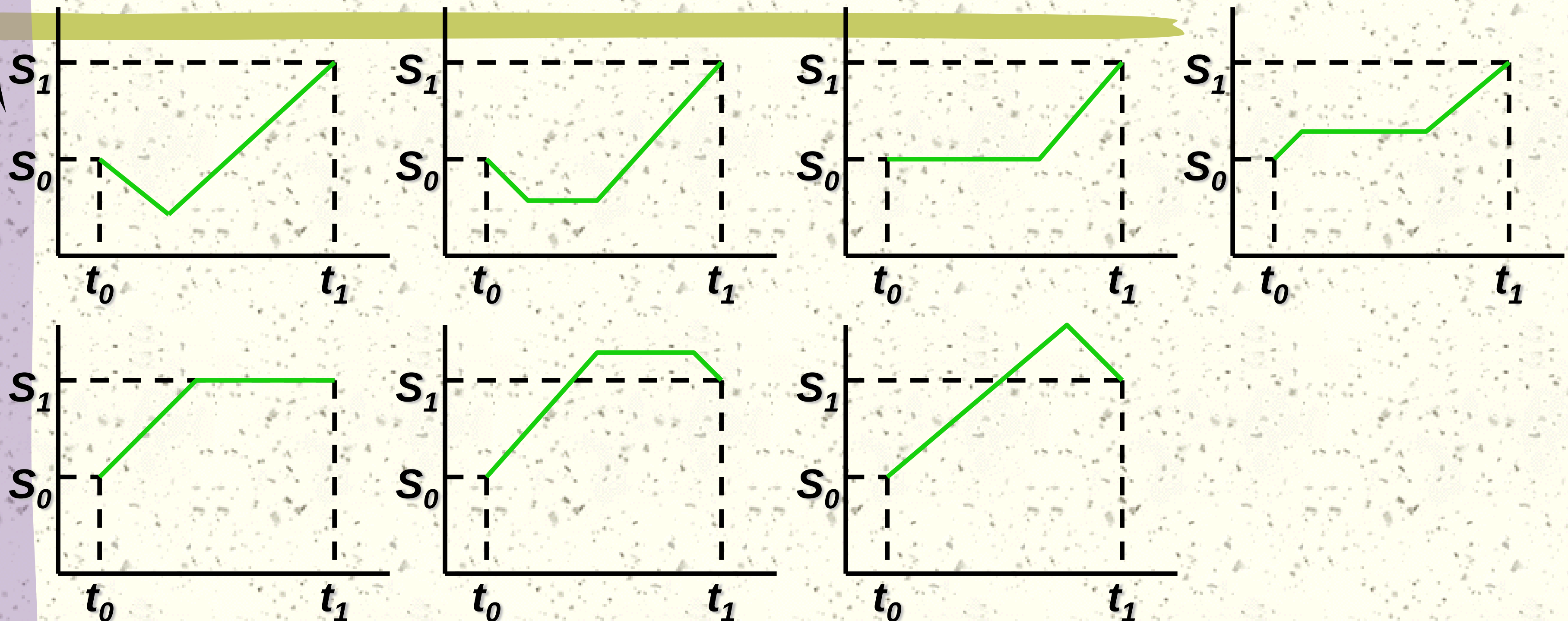
2005: parallelism on multi-processor (ASPDAC)

2006: dual-processor fault-tolerant system (ASAP)

2007: leakage aware DVS (AHS), multi-core system scheduling (McSoC)

2013: temperature-aware DVS (ASAP)

Solution: Feasible DVS System



- Change voltage only when necessary
- Change at the maximal rate
- Time(s) when voltage changes is calculable

L. Yuan and G. Qu. "What Is the Limit of Energy Saving by Dynamic Voltage Scaling", ICCAD'2001 .

Voltage Set-up Problem

- For a multiple-voltage DVS system to serve a set of applications $\{(e_i, d_i, p_i): i=1, 2, \ldots, n\}$ without missing their deadlines, where e_i: execution time d_i: deadline, p_i: probability d_i occurs.
  - if the system has m voltages $\{v_1, v_2, \ldots, v_m\}$, determine the value of each v_i to minimize the average energy consumption.
  - determine m and the value of each v_i.

S. Hua and G. Qu. "Approaching the Maximum Energy Saving on Embedded Systems with Multiple Voltages", ICCAD'2003 .

Information on Two Applications

Application	Deadline	Execution Time	Probability	V_i^0 (V)
A	10	9	0.03	3.0564
		4	0.18	1.8124
		3	0.39	1.5516
B	8	6	0.04	2.6888
		4	0.10	2.0669
		3	0.12	1.7479
		2	0.14	1.4176

V_{ref} = 3.3v

S. Hua and G. Qu. "Voltage Setup Problem for Embedded Systems with Multiple Voltages", TVLSI'2005 .

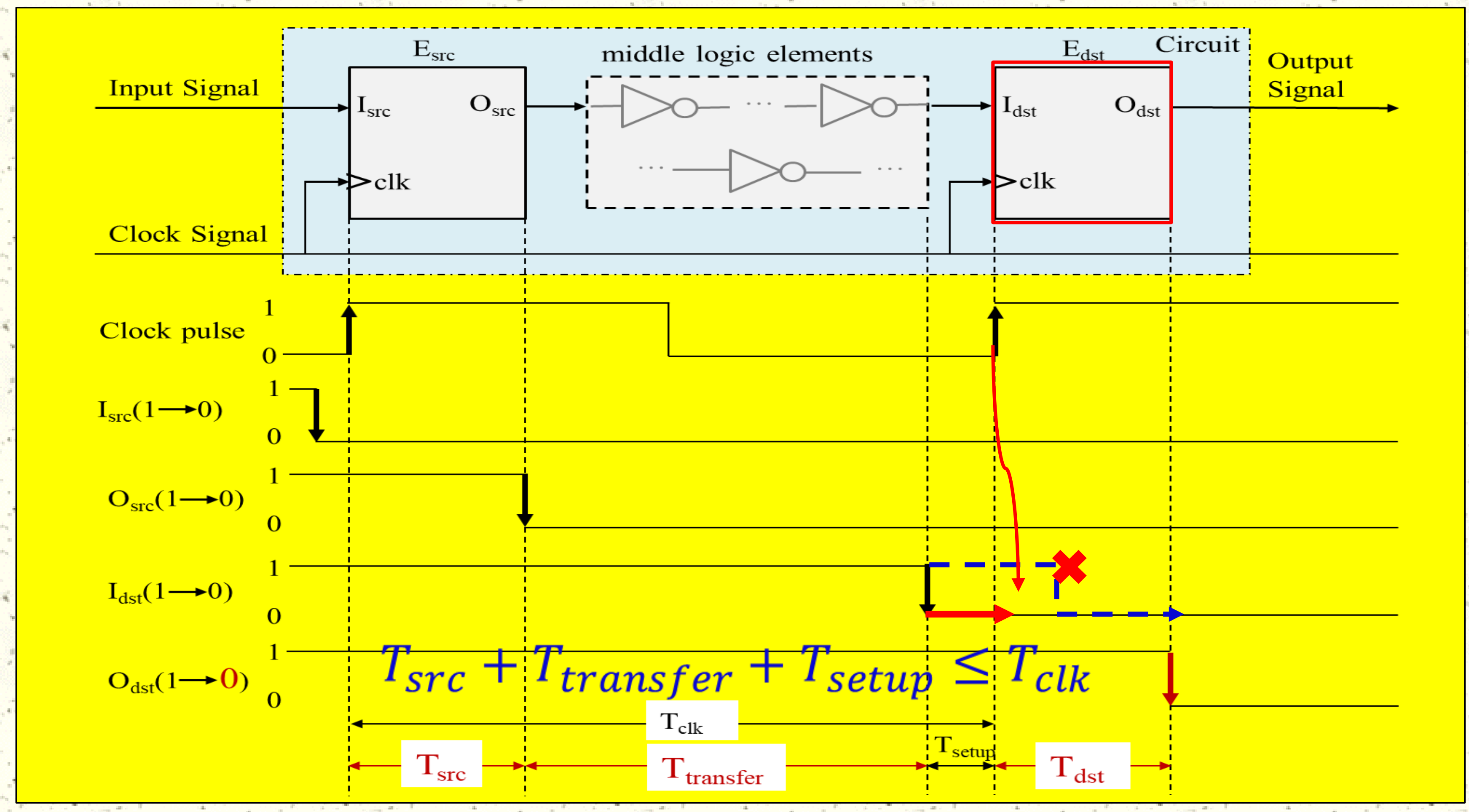
Reference Systems

DVS Systems	Voltages	Energy		
fixed-voltage	3.0564	2.9536		
ideal	--	1.1763		

DVS with Optimal Voltage Set-ups

DVS Systems	Voltages	Energy	vs. fixed-voltage	vs. Ideal
fixed-voltage	3.0564	2.9536	--	+151.1%
dual-voltage	3.0564 1.8124	1.3833	- 53.2%	+17.6%
3-voltage	3.0564 2.0688 1.5514	1.2337	- 58.2%	+4.9%
4-voltage	3.0564 2.0768 1.8119 1.5509	1.2071	- 59.1%	+2.6%
ideal	--	1.1763	--	--

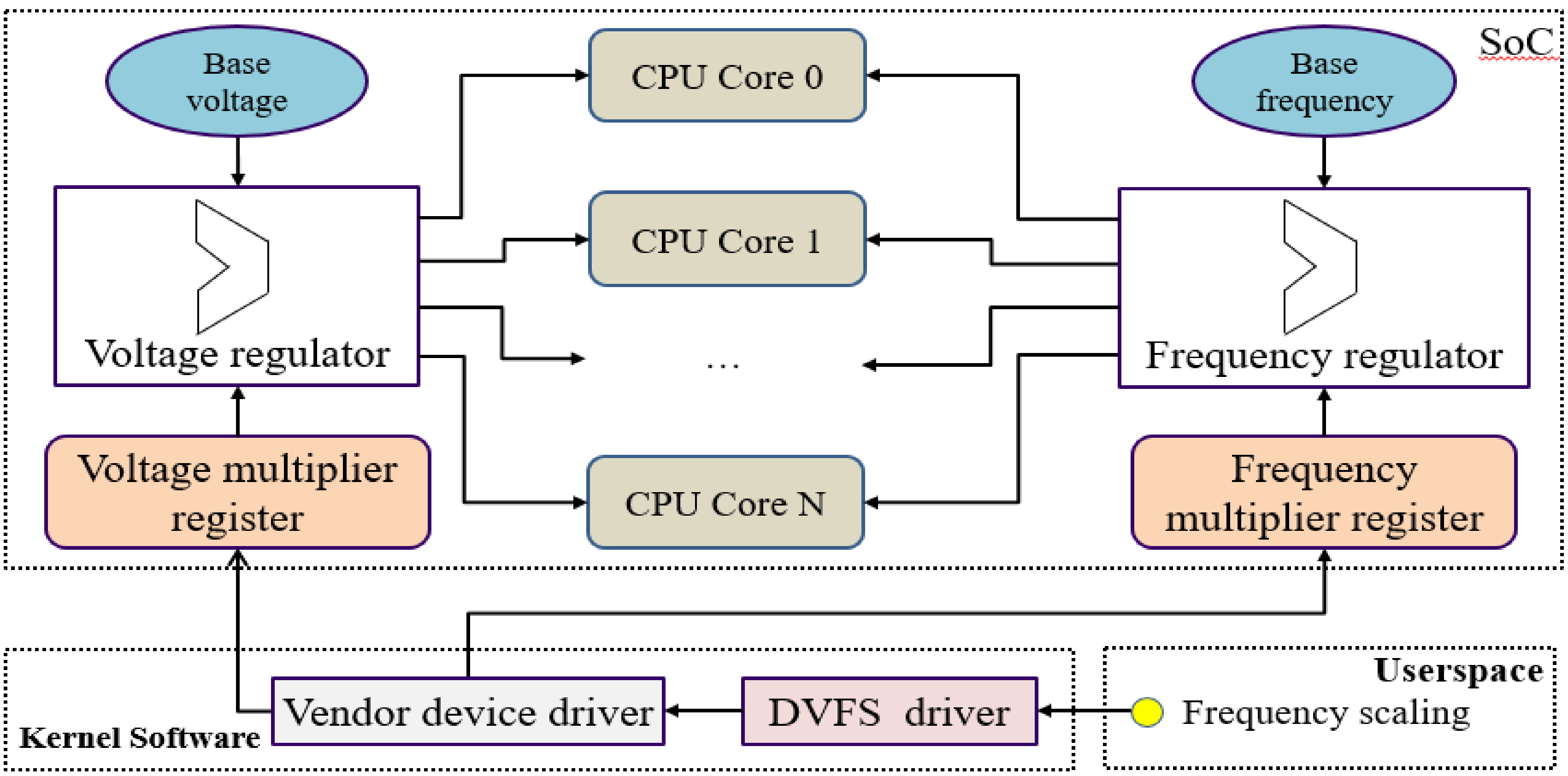
Circuit Timing Issues by DVS



$$T_{src} + T_{transfer} + T_{setup} \leq T_{clk}$$

P. Qiu, et al, "VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies", CCS'2019.

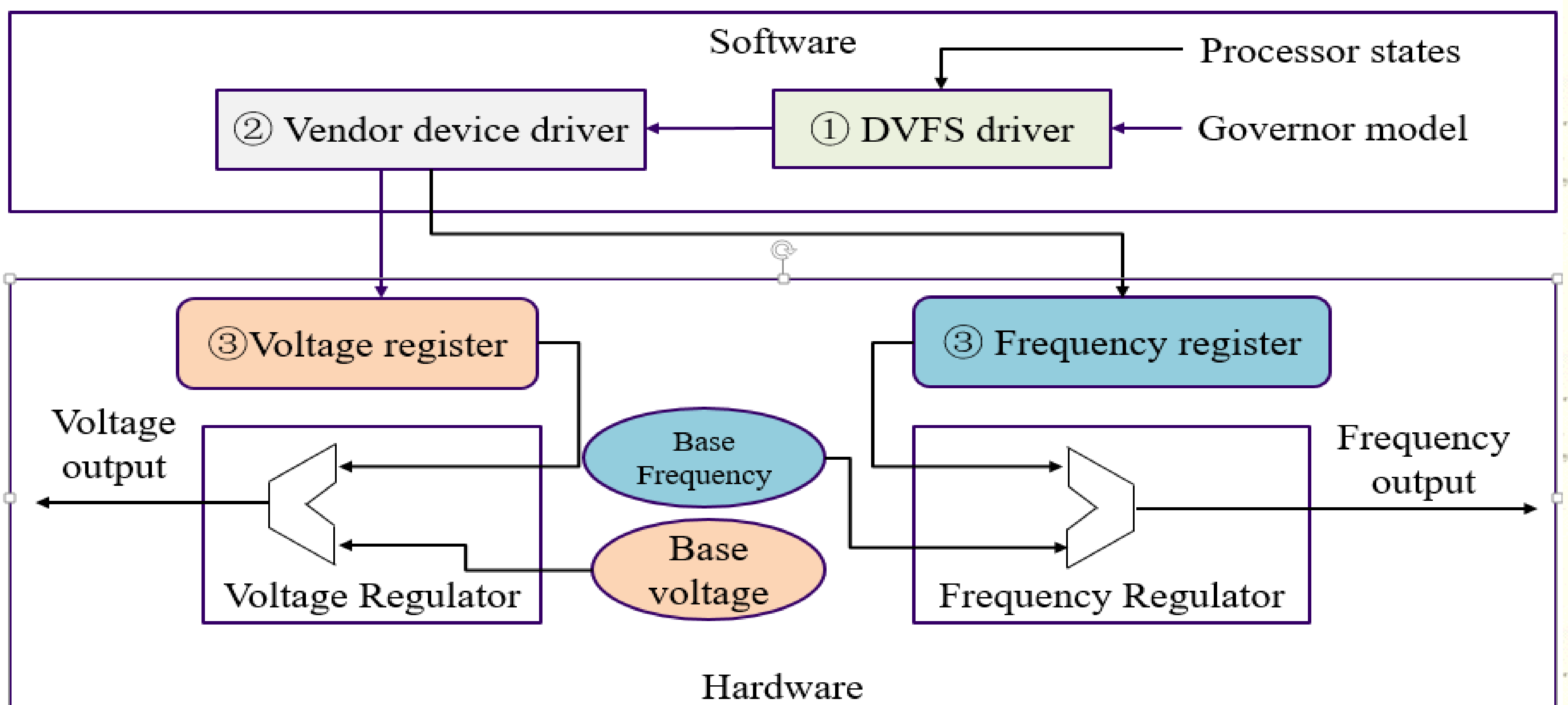
Multi-core DVFS Framework



- Ideal: each core has its own voltage and frequency
- Reality: all cores share the same V and F

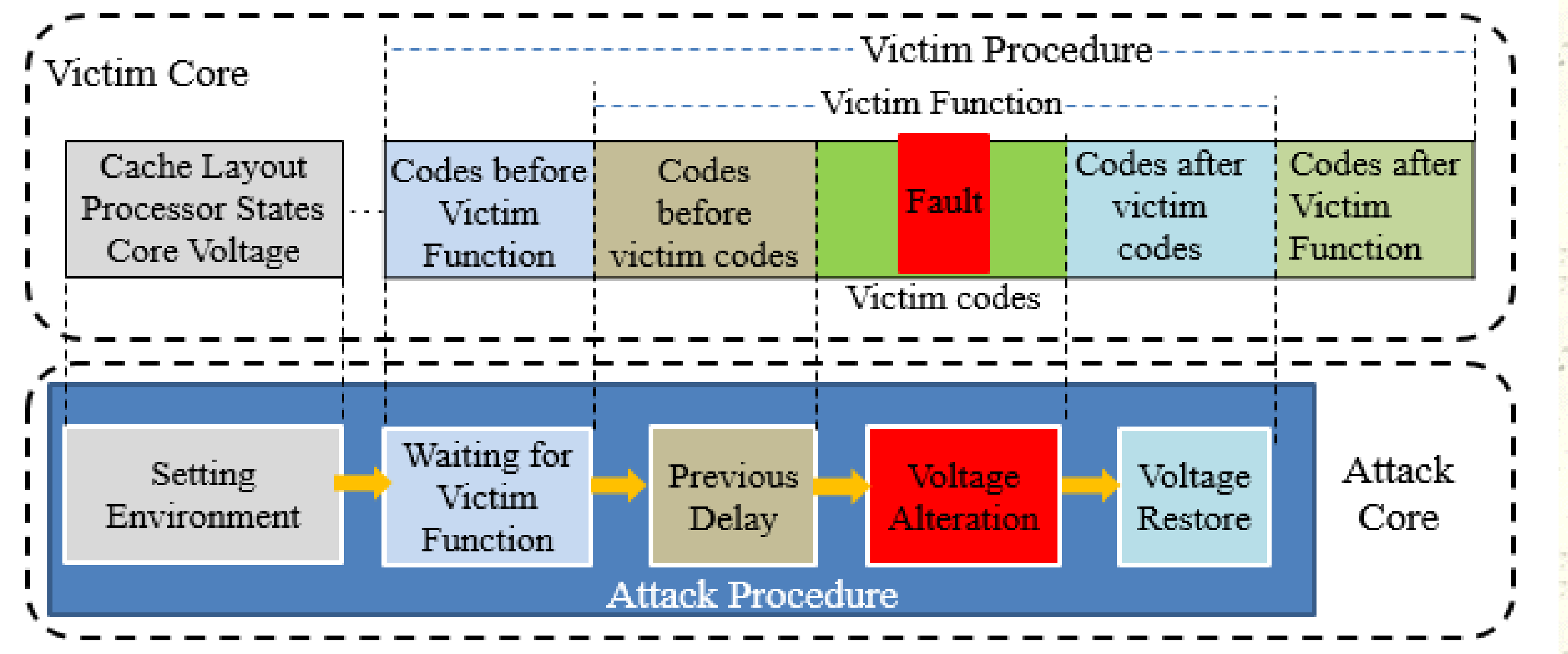
DVFS Working Flow

- DVFS driver selects proper V and F
- Vendor device driver changes V and F registers
- V and F registers alter the regulator outputs



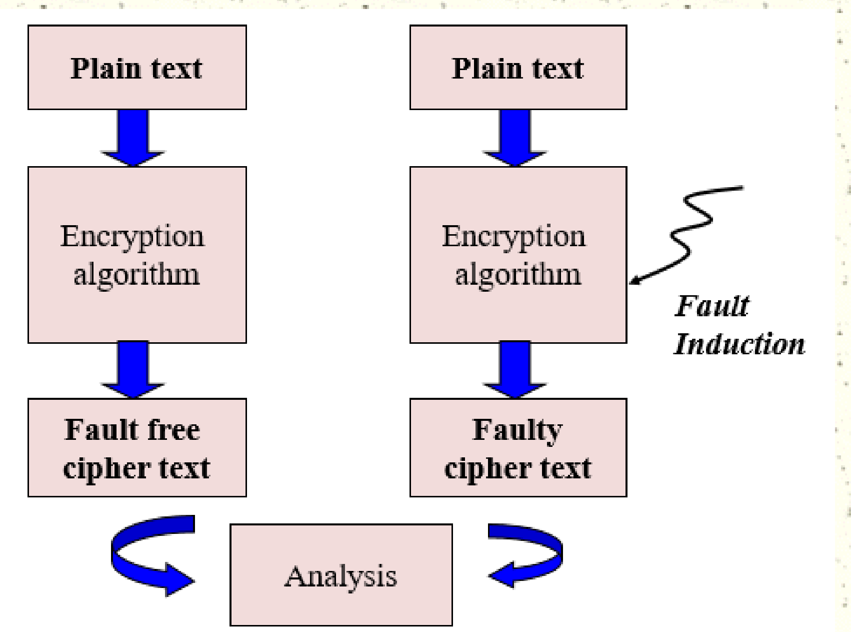
Overview of VoltJockey

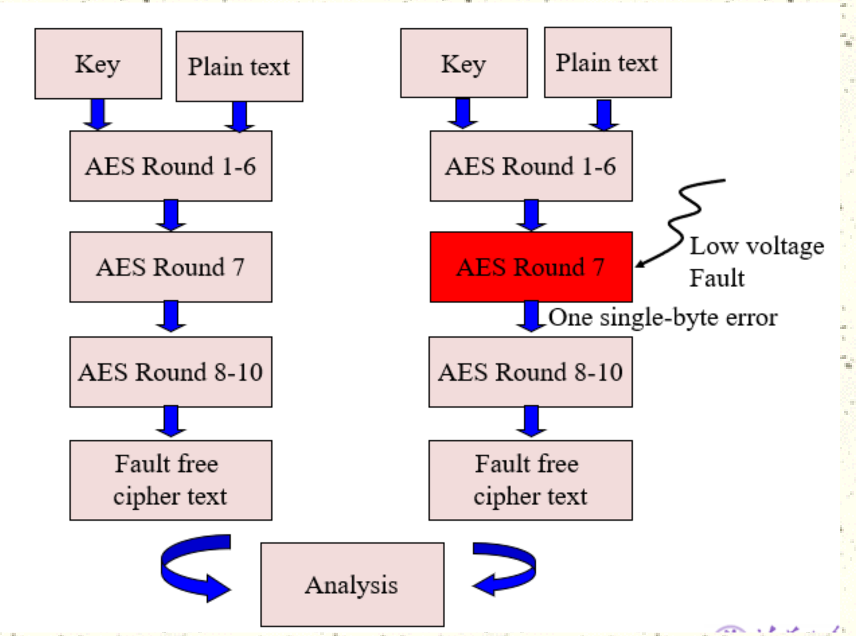
- The attacker procedure and victim procedure are executed on different cores.
- The victim core has a high frequency, but all the other cores have a low frequency.



Fault Injection Attacks

- DVFS is an effective way to generate faults
- The challenge is when and where to create the faults





Short History of VoltJockey

1 May 2017

Started the VoltJockey research

Sep. 2017: collaboration with UMD

2 February, 2018

Accomplished attack on **ARM TrustZone**

- *Feb. 2019*, Submitted paper to IEEE S&P
- *May. 2019*, Submitted paper to ACM CCS; DEMO on HOST

3 June, 2019

Accomplished attack on **Intel SGX**

- *June 2019*, Submitted paper to AsianHost
- *Aug. 2019*, Reported to Intel

4 Nov. 11, 2019

CCS 2019 published VoltJockey on ARM

5 December 2019

Intel confirmed VoltJockey & published CVE-2019-11157

- AisianHost 2019 be the first official venue to publish the paper

......

VoltJockey

http://cpu.cs.tsinghua.edu.cn/

VoltJockey

Impact

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data.

Vulnerability Scoring Details

CVE	Score
CVE-2019-11157	7.9 (HIGH)

VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies

Pengfei Qiu[1,2,3], Dongsheng Wang[1,2], Yongqiang Lyu[2*], Gang Qu[3]

We validate VoltJockey on an ARM-based *Krait* processor by breaking AES and RSA in TrustZone. The experiments successfully obtain the encryption key of AES and load untrusted applications into TrustZone by invalidating the RSA verification.

VoltJockey

VoltJockey

Impact

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data.

Vulnerability Scoring Details

CVE	Score
CVE-2019-11157	7.9 (HIGH)

VoltJockey: Breaking SGX by Software-Controlled Voltage-Induced Hardware Faults

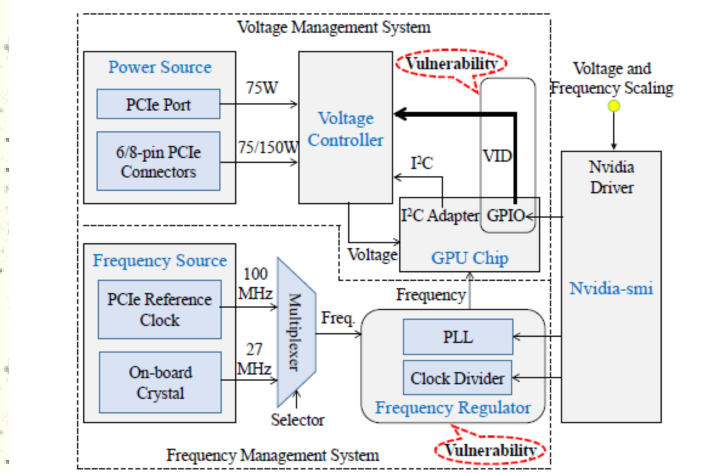
Pengfei Qiu[1,2,3], Dongsheng Wang[1,2], Yongqiang Lyu[2*], Gang Qu[3]

2) We propose a hardware fault attack based on our developed kernel module. To the best of our knowledge, unlike the existing attacks on SGX, this is the first fault injection attack that does not rely on any software vulnerability.
3) We apply the proposed attack on a commercial Intel processor with AES running in the enclave and successfully obtain the encryption key.

intel®

Lightning

Lightning: Striking the Secure Isolation on GPU Clouds with Transient Hardware Faults



- We propose the *Lightning*, the method based on DVFS faults which not only degrades model accuracy, but also leads the model to misclassify inputs to our desired inference output (targeted attack).
- We verify the method on three commodity Nvidia GPUs and show that *Lightning* can reduce CNN accuracy on MNIST, CIFAR-10, and Yale face data sets by 64.5% on average, and achieves a 67.9% success rate for the targeted attack on Lenet-5 model.

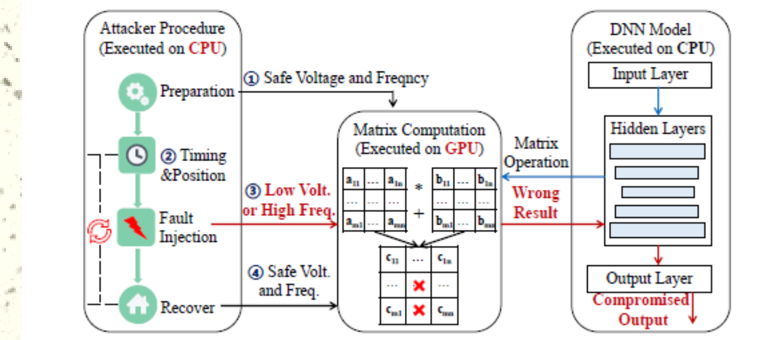
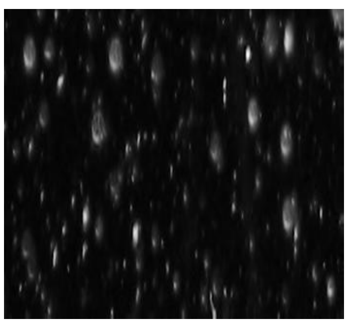
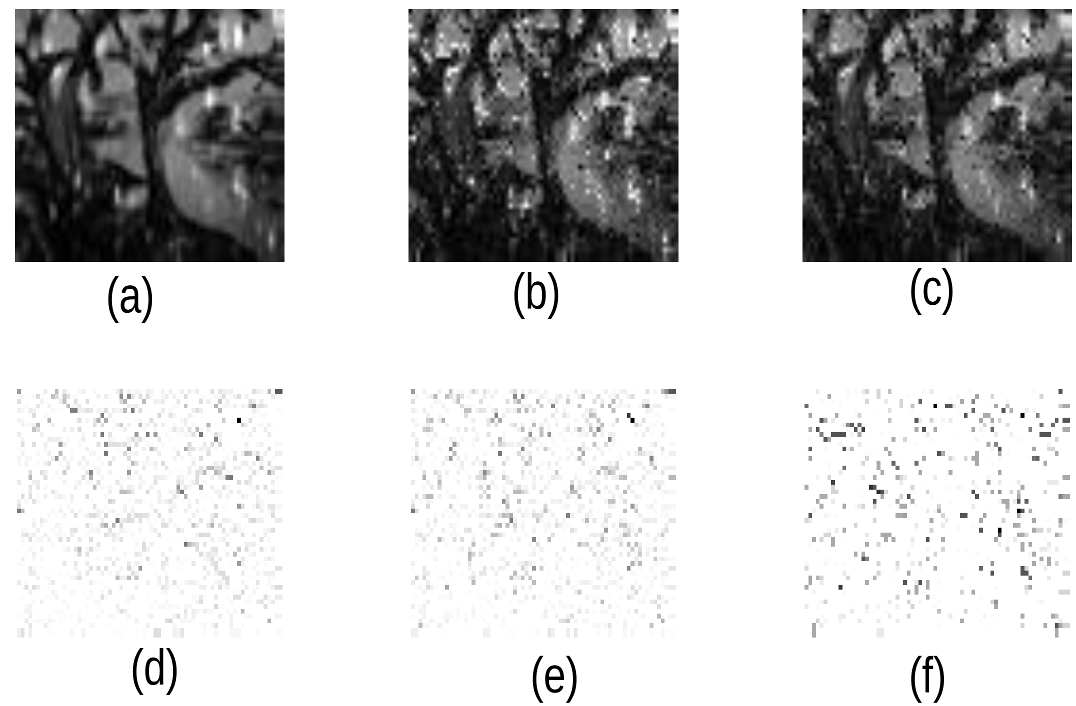


Figure 3: The exploitation of the DVFS-related defects. The exploitation procedure takes four steps to complete the process: ① configure CPU and GPU with a safe voltage and frequency; ② wait for the fault injection points; ③ create low-voltage or high-frequency glitches to induce faults into the GPU; ④ recover the safe voltage and frequency for the GPU.

DVS for Device Authentication

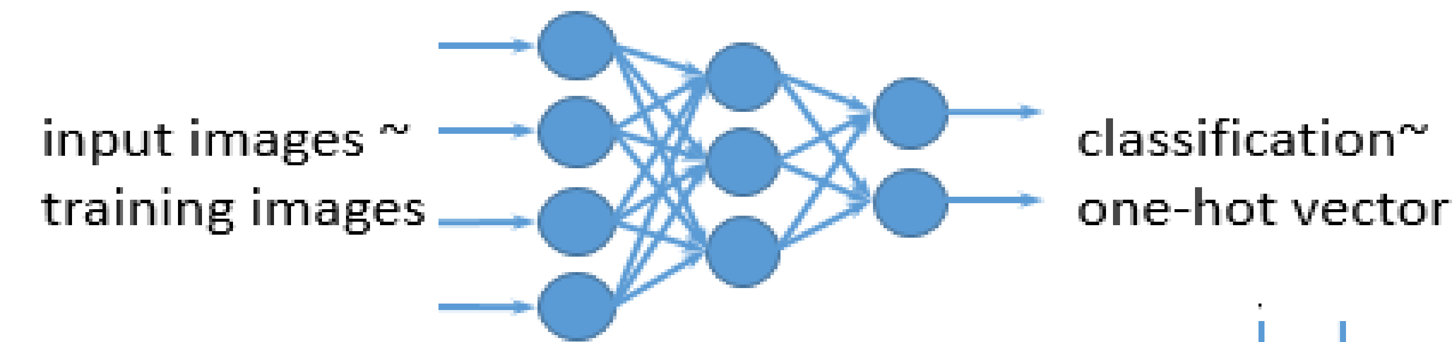




MeshSec Lab

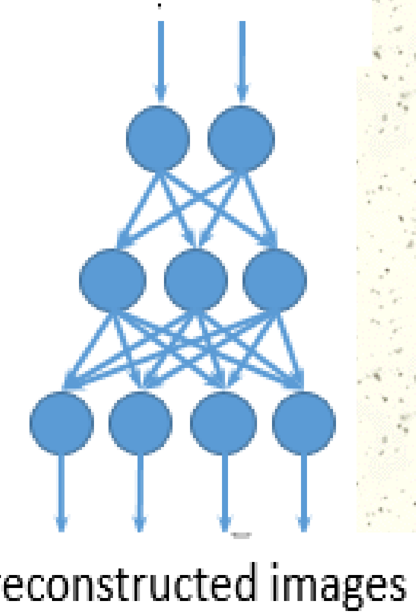
M.T. Arafin, M. Gao, and G. Qu, "VOLtA: Voltage Over-scaling Based Lightweight Authentication for IoT Applications", ASPDAC'2017.

What is Model Inversion Attack?



Algorithm 1 Inversion attack for facial recognition models.

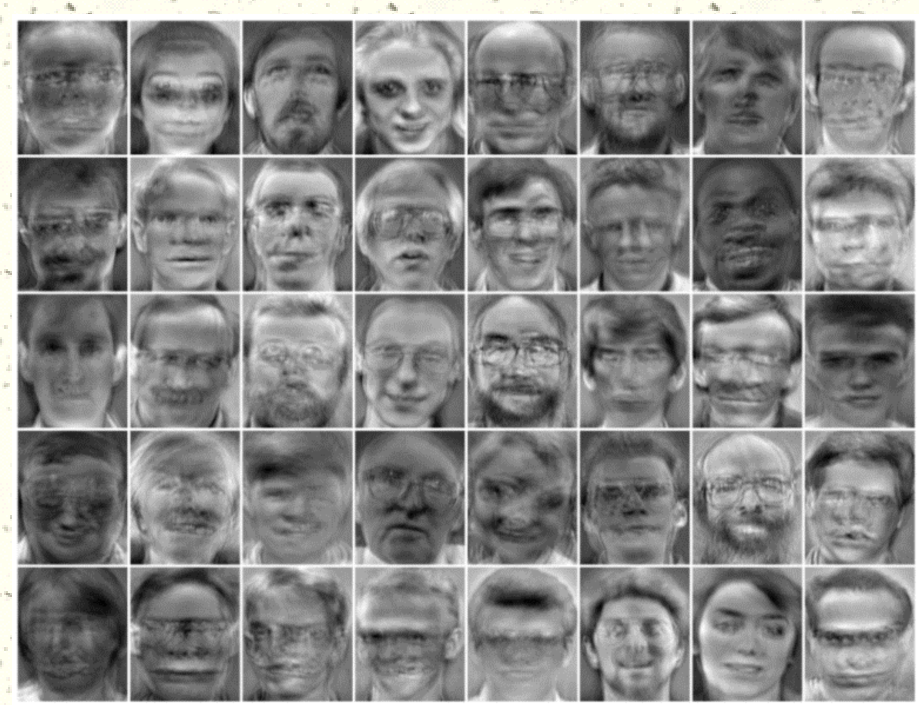
function MI-FACE($label, \alpha, \beta, \gamma, \lambda$)
  $c(\mathbf{x}) \stackrel{\text{def}}{=} 1 - \tilde{f}_{label}(\mathbf{x}) + \text{AUXTERM}(\mathbf{x})$
  $\mathbf{x}_0 \leftarrow \mathbf{0}$
  for $i \leftarrow 1 \ldots \alpha$ **do**
    $\mathbf{x}_i \leftarrow \text{PROCESS}(\mathbf{x}_{i-1} - \lambda \cdot \nabla c(\mathbf{x}_{i-1}))$
    if $c(\mathbf{x}_i) \geq \max(c(\mathbf{x}_{i-1}), \ldots, c(\mathbf{x}_{i-\beta}))$ **then**
      break
    if $c(\mathbf{x}_i) \leq \gamma$ **then**
      break
  return $[\arg\min_{\mathbf{x}_i}(c(\mathbf{x}_i)), \min_{\mathbf{x}_i}(c(\mathbf{x}_i))]$



M. Fredrikson et al, Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures, CCS, 2015.

What is Model Inversion Attack?

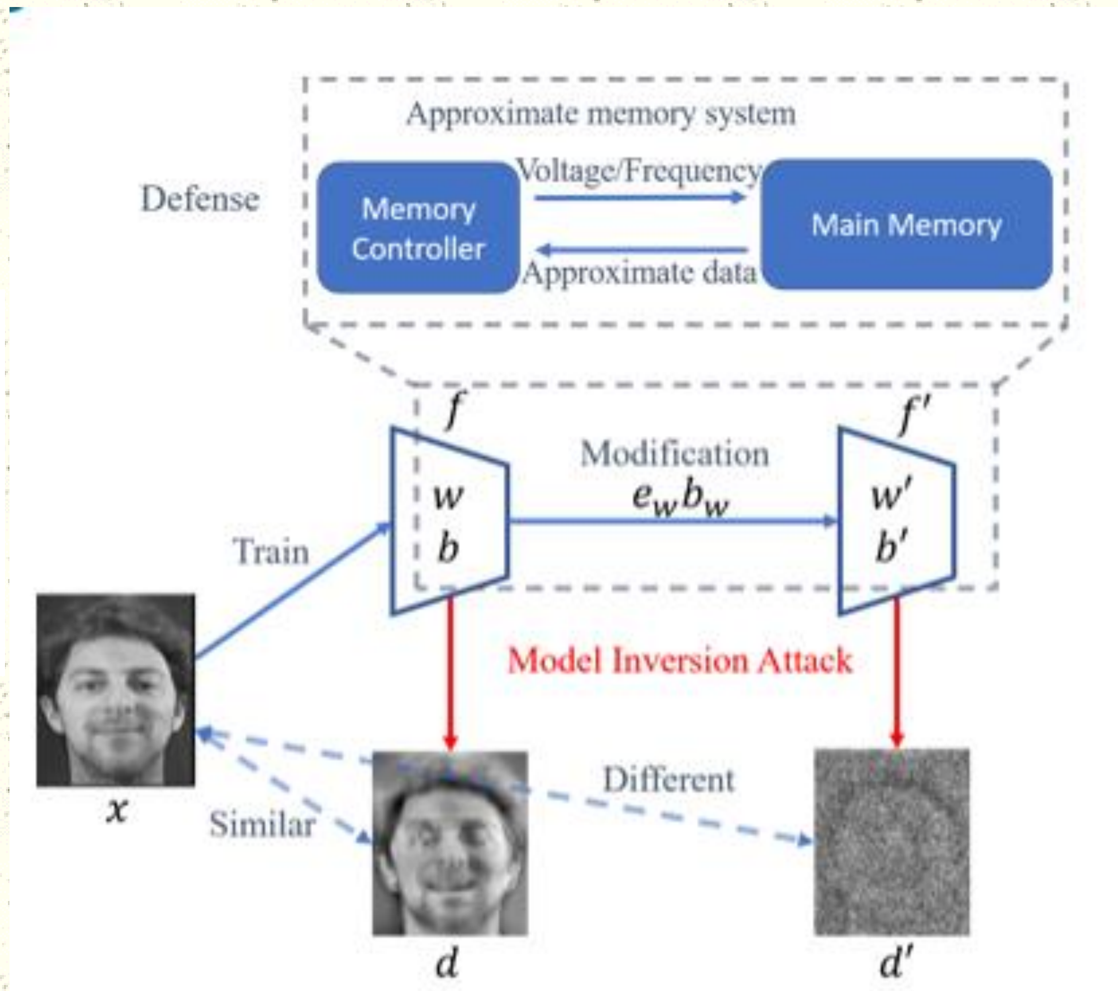
- Training data: b&w images of 40 people.



M.T. Arafin, Q. Xu, and G. Qu, "MIDAS: Model Inversion Defenses using an Approximate Memory System", AsianHOST'2020.

MIDAS Approach

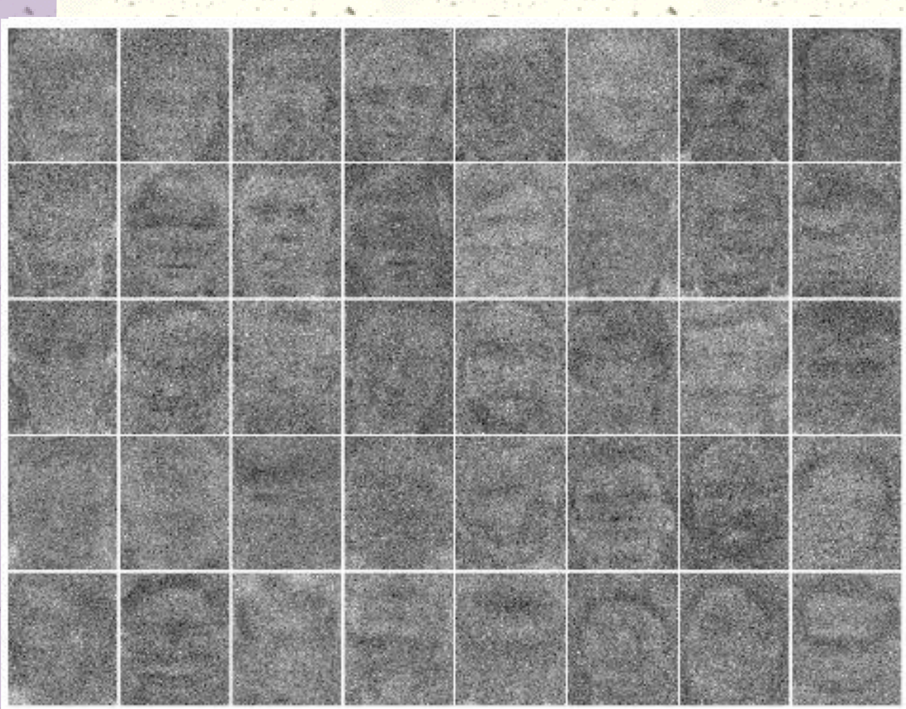
- Approximate memory system



M.T. Arafin, Q. Xu, and G. Qu, “MIDAS: Model Inversion Defenses using an Approximate Memory System”, AsianHOST’2020.

Protection with MIDAS

- Training data: b&w images of 40 people.



M.T. Arafin, Q. Xu, and G. Qu, "MIDAS: Model Inversion Defenses using an Approximate Memory System", AsianHOST'2020.

Conclusion

DVFS will evolve, but will not die

- More applications, devices, greedy human nature → higher power/energy demand
- Security and privacy are emerging
  - CLKscrew, Plundervolt, VOLTpwn
  - cover channel (DVFSspy)
  - side-channel attacks (PLATYPUS), …
- Holistic approach is needed:
  - Circuit, memory, architecture, OS, application, networking, human, …

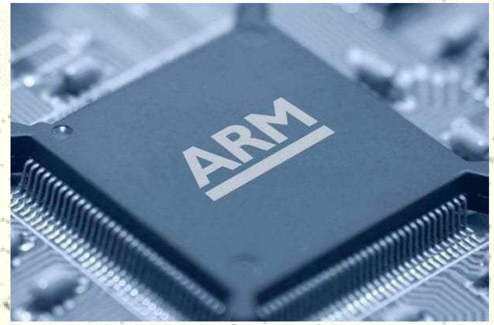
VoltJockey + Lightning

Impact

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data.

Vulnerability Scoring Details

CVE	Score
CVE-2019-11157	7.9 (HIGH)



P. Qiu, D. Wang, Y. Lyu, and G. Qu, "VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies", CCS'2019.

P. Qiu, D. Wang, Y. Lyu, and G. Qu, "VoltJockey: Breaking SGX by Software-Controlled Voltage-Induced Hardware Faults", AsianHOST'2019. (Best paper award)