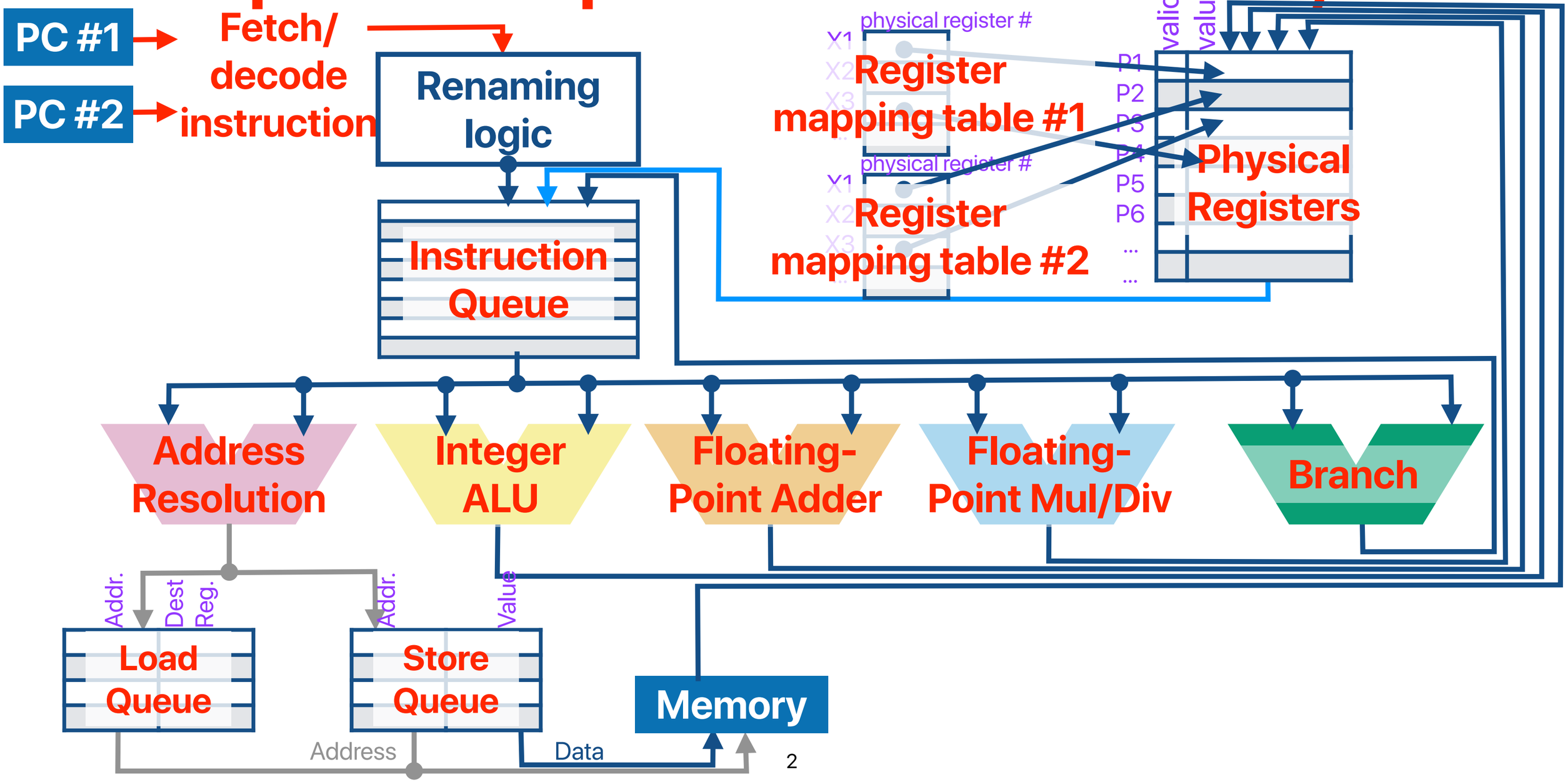
Dark Silicon & Modern Computer Architecture

Hung-Wei Tseng

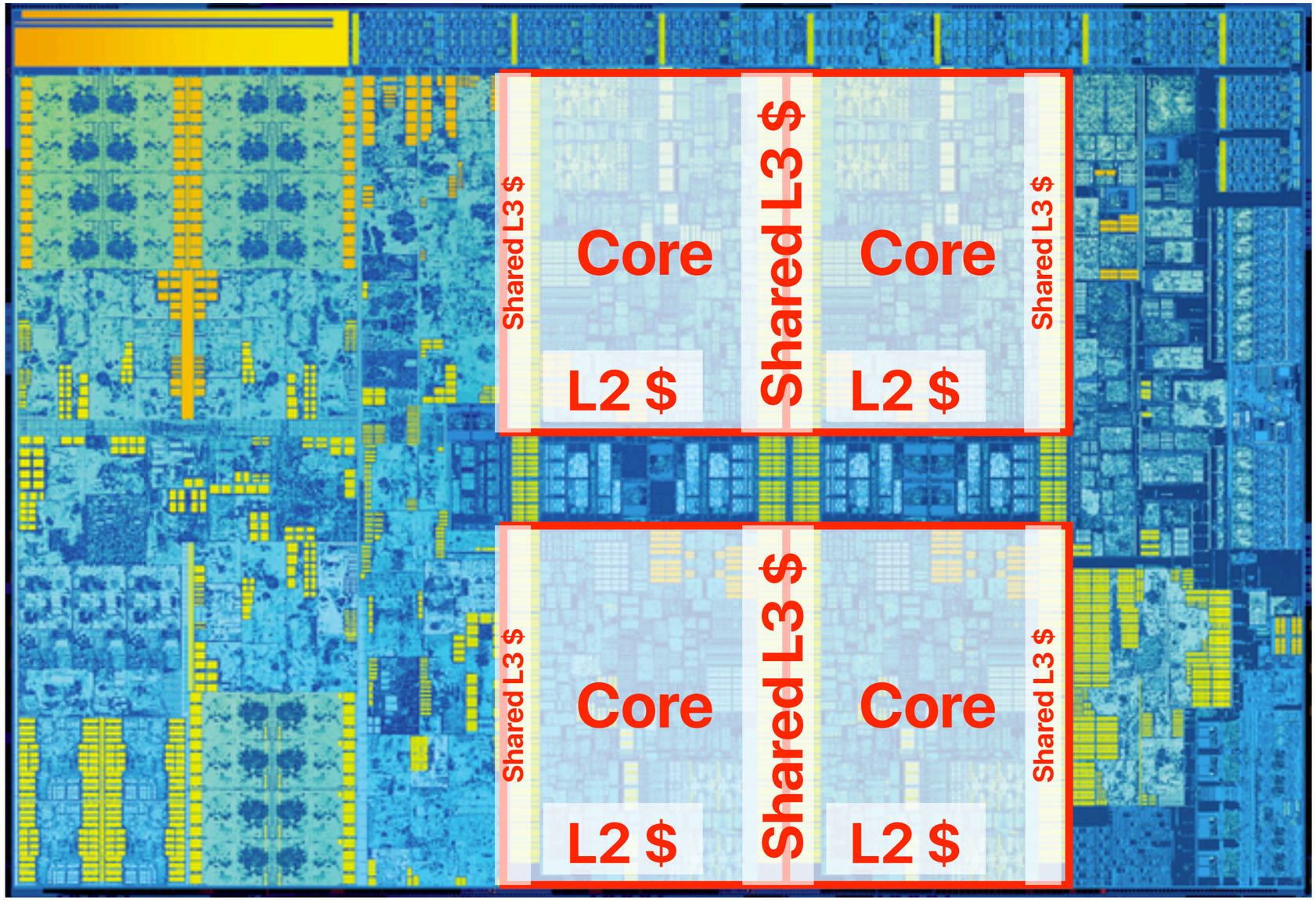
Recap: SMT SuperScalar Processor w/ ROB



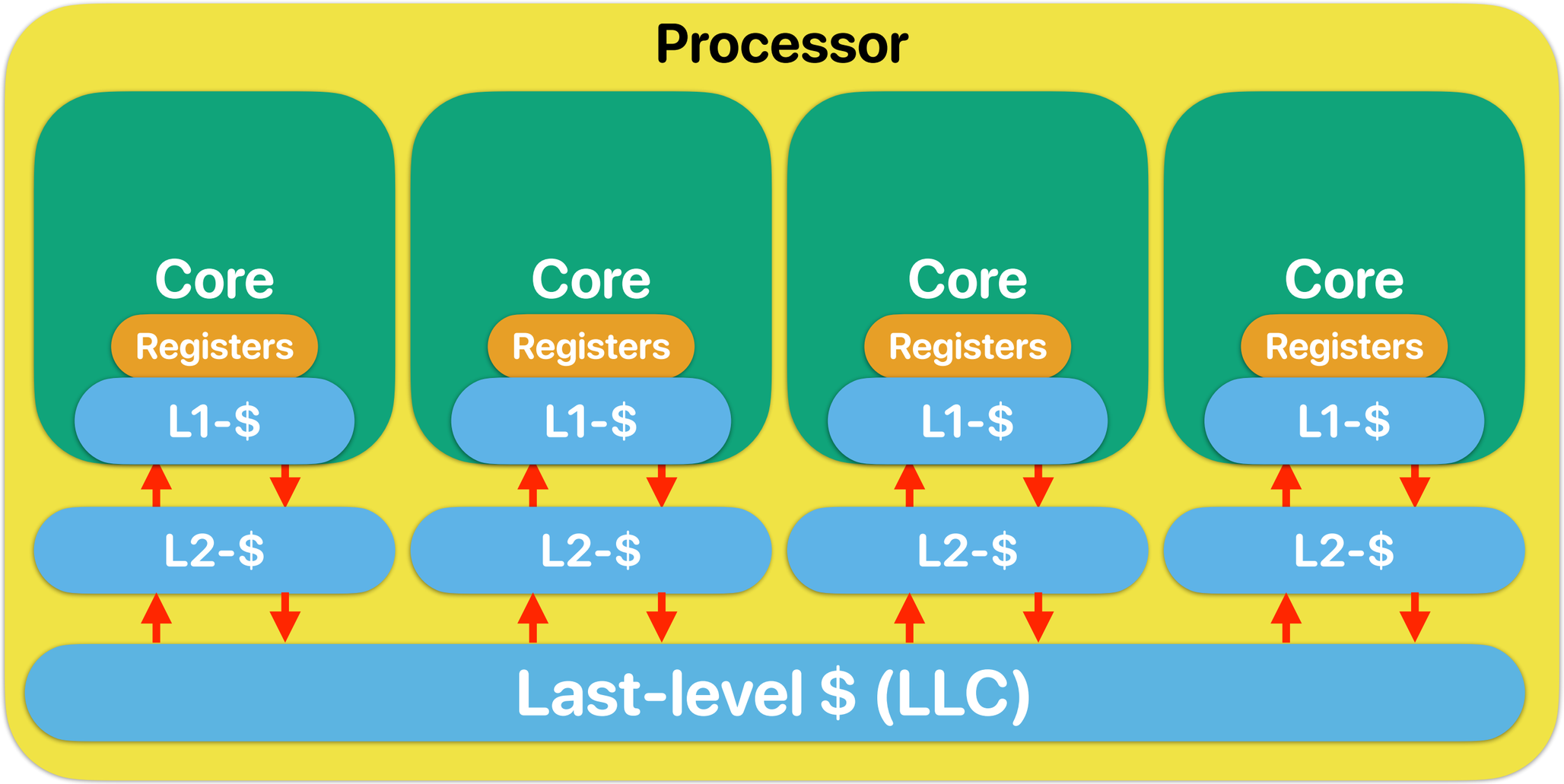
Recap: SMT

- Improve the throughput of execution
  - May increase the latency of a single thread
- Less branch penalty per thread
- Increase hardware utilization
- Simple hardware design: Only need to duplicate PC/Register Files — the pipeline/L1-cache are **shared** between hardware threads in the same core
- Real Case:
  - Intel HyperThreading (supports up to two threads per core)
    - Intel Pentium 4, Intel Atom, Intel Core i7
  - AMD RyZen

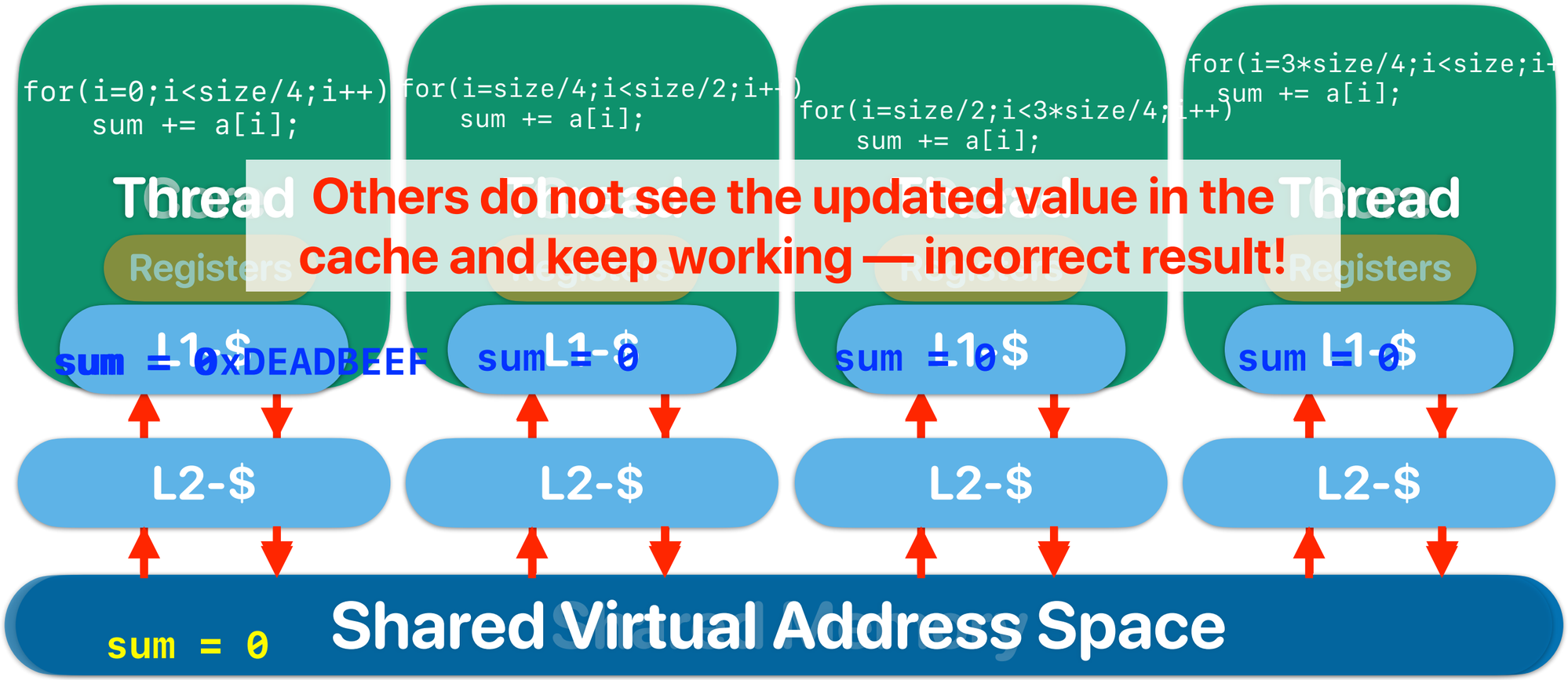
Recap: Intel SkyLake



Recap: Concept of CMP



Recap: What software thinks about “multiprogramming” hardware



Others do not see the updated value in the cache and keep working — incorrect result!

Performance comparison

- Comparing implementations of thread_vadd — L and R, please identify which one will be performing better and why

Version L

```
void *threaded_vadd(void *thread_id)
{
  int tid = *(int *)thread_id;
  int i;
  for(i=tid;i<ARRAY_SIZE;i+=NUM_OF_THREADS)
  {
        c[i] = a[i] + b[i];
  }
  return NULL;
}
```

Version R

```
void *threaded_vadd(void *thread_id)
{
  int tid = *(int *)thread_id;
  int i;
  for(i=tid*(ARRAY_SIZE/NUM_OF_THREADS);i<(tid+1)*(ARRAY_SIZE/NUM_OF_THREADS);i++)
  {
      c[i] = a[i] + b[i];
  }
  return NULL;
}
```

A. L is better, because the cache miss rate is lower
B. R is better, because the cache miss rate is lower
C. L is better, because the instruction count is lower
D. R is better, because the instruction count is lower
E. Both are about the same

Main thread

```
for(i = 0 ; i < NUM_OF_THREADS ; i++)
{
  tids[i] = i;
  pthread_create(&thread[i], NULL, threaded_vadd, &tids
}
for(i = 0 ; i < NUM_OF_THREADS ; i++)
  pthread_join(thread[i], NULL);
```

L v.s. R

Version L

```
void *threaded_vadd(void *thread_id)
{
  int tid = *(int *)thread_id;
  int i;
  for(i=tid;i<ARRAY_SIZE;i+=NUM_OF_THREADS)
  {
        c[i] = a[i] + b[i];
  }
  return NULL;
}
```

Version R

```
void *threaded_vadd(void *thread_id)
{
  int tid = *(int *)thread_id;
  int i;
  for(i=tid*(ARRAY_SIZE/NUM_OF_THREADS);i<(tid+1)*(ARRAY_SIZE/NUM_OF_THREADS);i++)
  {
      c[i] = a[i] + b[i];
  }
  return NULL;
}
```

C 

C 

Coherency & Consistency

- Coherency — Guarantees all processors see the same value for a variable/memory address in the system when the processors need the value at the same time
  - What value should be seen
- Consistency — All threads see the change of data in the same order
  - When the memory operation should be done

Team scores

8	16	14	11

Outline

- Parallel programming (cont.)
- Power and energy
- Dark Silicon and its impact on computer architecture

Again — how many values are possible?

- Consider the given program. You can safely assume the caches are coherent. How many of the following outputs will you see?

① (0, 0)
② (0, 1)
③ (1, 0)
④ (1, 1)

A. 0
B. 1
C. 2
D. 3
E. 4

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

volatile int a,b;
volatile int x,y;
volatile int f;
void* modifya(void *z) {
  a=1;
  x=b;
  return NULL;
}
void* modifyb(void *z) {
  b=1;
  y=a;
  return NULL;
}
```

```c
int main() {
  int i;
  pthread_t thread[2];
  pthread_create(&thread[0], NULL, modifya, NULL);
  pthread_create(&thread[1], NULL, modifyb, NULL);
  pthread_join(thread[0], NULL);
  pthread_join(thread[1], NULL);
  fprintf(stderr,"(%d, %d)\n",x,y);
  return 0;
}
```

Possible scenarios

Thread 1	Thread 2
a=1;	b=1;
x=b;	y=a;

(1,1)

Thread 1	Thread 2
a=1;	b=1;
x=b;	y=a;

(0,1)

Thread 1	Thread 2
a=1;	b=1;
x=b;	y=a;

(1,0)

Thread 1	Thread 2
x=b;	y=a;
a=1;	b=1;

OoO Scheduling!

(0,0)

Why (0,0)?

- Processor/compiler may reorder your memory operations/ instructions
  - Coherence protocol can only guarantee the update of the same memory address
  - Processor can serve memory requests without cache miss first
  - Compiler may store values in registers and perform memory operations later
- Each processor core may not run at the same speed (cache misses, branch mis-prediction, I/O, voltage scaling and etc..)
- Threads may not be executed/scheduled right after it's spawned

fence instructions

- x86 provides an "mfence" instruction to prevent reordering across the fence instruction
- x86 only supports this kind of "relaxed consistency" model. You still have to be careful enough to make sure that your code behaves as you expected

thread 1	thread 2

`a=1;`
`mfence` **a=1 must occur/update before mfence**
`x=b;`
	`b=1;`
	`mfence` **b=1 must occur/update before mfence**
	`y=a;`

Take-aways of parallel programming

- Processor behaviors are non-deterministic
  - You cannot predict which processor is going faster
  - You cannot predict when OS is going to schedule your thread
- Cache coherency only guarantees that everyone would eventually have a coherent view of data, but not when
- Cache consistency is hard to support

Power and Energy

Poll close in 1:30

Power & Energy

- Regarding power and energy, how many of the following statements are correct?
  - ① Lowering the power consumption helps reducing the heat generation
  - ② Lowering the energy consumption helps reducing the electricity bill
  - ③ Lowering the power consumption helps extending the battery life
  - ④ A CPU with 10% utilization can still consume 33% of the peak power
  - A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

Poll close in 1:30

Power & Energy

- Regarding power and energy, how many of the following statements are correct?
  - ① Lowering the power consumption helps reducing the heat generation
  - ② Lowering the energy consumption helps reducing the electricity bill
  - ③ Lowering the power consumption helps extending the battery life
  - ④ A CPU with 10% utilization can still consume 33% of the peak power
  - A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

Power v.s. Energy

- Power is the direct contributor of “heat”
  - Packaging of the chip
  - Heat dissipation cost
  - Power = $P_{Dynamic} + P_{static}$
- Energy = P * ET
  - The electricity bill and battery life is related to energy!
  - Lower power does not necessary means better battery life if the processor slow down the application too much

Dynamic Power

Dynamic/Active Power

- The power consumption due to the switching of transistor states
- Dynamic power per transistor

$$P_{dynamic} \sim \alpha \times C \times V^2 \times f \times N$$

  - α: average switches per cycle
  - C: capacitance
  - V: voltage
  - f: frequency, usually linear with V
  - N: the number of transistors

Double Clock Rate or Double the # of Processors?

- Assume 60% of the application can be fully parallelized with 2-core or speedup linearly with clock rate. Should we **double the clock rate** or **duplicate a core**?

$$P_{dynamic} \sim \alpha \times C \times V^2 \times f \times N$$

$$Speedup_{parallel}(f_{parallelizable}, n) = \frac{1}{(1 - f_{parallelizable}) + \frac{f\_parallelizable}{n}}$$

$$Speedup_{parallel}(60\,\%, 2) = \frac{1}{(1 - 60\%) + \frac{60\,\%}{2}} = 1.43$$

$$Power_{2-core} = 2 \times P_{baseline}$$

$$Energy_{2-core} = 2 \times P_{baseline} \times ET_{baseline} \times \frac{1}{1.43} = 1.39 \times Energy_{baseline}$$

$$Speedup_{2\times clock} = 2$$

$$Power_{2\times clock} = 2^3 \times P_{baseline} = 8 \times P_{baseline}$$

$$Energy_{2\times clock} = 2^3 \times P_{baseline} \times ET_{baseline} \times \frac{1}{2} = 4 \times P_{baseline} \times ET_{baseline}$$

Dynamic voltage/frequency scaling

- Dynamically lower power for performance
  - Change the voltage and frequency at runtime
  - Under control of operating system — that's why updating iOS may slow down an old iPhone
- Recall: $P_{dynamic} \sim \alpha \times C \times V^2 \times f \times N$
  - Because frequency ~ to V…
  - $P_{dynamic}$ ~ to V^3
- Reduce both V and f linearly
  - Cubic decrease in dynamic power
  - Linear decrease in performance (actually sub-linear)
    - Thus, only about quadratic in energy
  - Linear decrease in static power
    - Thus, only modest static energy improvement
  - Newer chips can do this on a per-core basis
    - cat /proc/cpuinfo in linux

Demo — changing the max frequency and performance

- Change the maximum frequency of the intel processor — you learned how to do this when we discuss programmer's impact on performance
- LIKWID a profiling tool providing power/energy information
  - likwid-perfctr -g ENERGY [command_line]
  - Let's try blockmm and popcount and see what's happening!

Power & Energy

- Regarding power and energy, how many of the following statements are correct?
  - ① Lowering the power consumption helps reducing the heat generation
  - ② Lowering the energy consumption helps reducing the electricity bill
  - ③ Lowering the power consumption helps extending the battery life
  - ④ A CPU with 10% utilization can still consume 33% of the peak power
  - A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

Poll close in 1:30

What happens if power doesn't scale with process technologies?

- If we are able to cram more transistors within the same chip area (Moore's law continues), but the power consumption per transistor remains the same. Right now, if put more transistors in the same area because the technology allows us to. How many of the following statements are true?
  - ① The power consumption per chip will increase
  - ② The power density of the chip will increase
  - ③ Given the same power budget, we may not able to power on all chip area if we maintain the same clock rate
  - ④ Given the same power budget, we may have to lower the clock rate of circuits to power on all chip area

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

What happens if power doesn't scale with process technol

- If we are able to cram more transistors within the same chip area (Moore's law continues), but the power consumption per transistor remains the same. Right now, if put more transistors in the same area because the technology allows us to. How many of the following statements are true?
  - ① The power consumption per chip will increase
  - ② The power density of the chip will increase
  - ③ Given the same power budget, we may not able to power on all chip area if we maintain the same clock rate
  - ④ Given the same power budget, we may have to lower the clock rate of circuits to power on all chip area

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

What happens if power doesn't scale with process technologies?

- If we are able to cram more transistors within the same chip area (Moore's law continues), but the power consumption per transistor remains the same. Right now, if put more transistors in the same area because the technology allows us to. How many of the following statements are true?
  - ① The power consumption per chip will increase
  - ② The power density of the chip will increase
  - ③ Given the same power budget, we may not able to power on all chip area if we maintain the same clock rate
  - ④ Given the same power budget, we may have to lower the clock rate of circuits to power on all chip area

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Dark Silicon and the End of Multicore Scaling

H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam and D. Burger
University of Washington, University of Wisconsin—Madison, University of Texas at Austin, Microsoft Research

Static/Leakage Power

- The power consumption due to leakage — transistors do not turn all the way off during no operation
- Becomes the **dominant** factor in the most advanced process technologies.

$$P_{leakage} \sim N \times V \times e^{-V_t}$$

  - N: number of transistors
  - V: voltage
  - V_t: threshold voltage where transistor conducts (begins to switch)

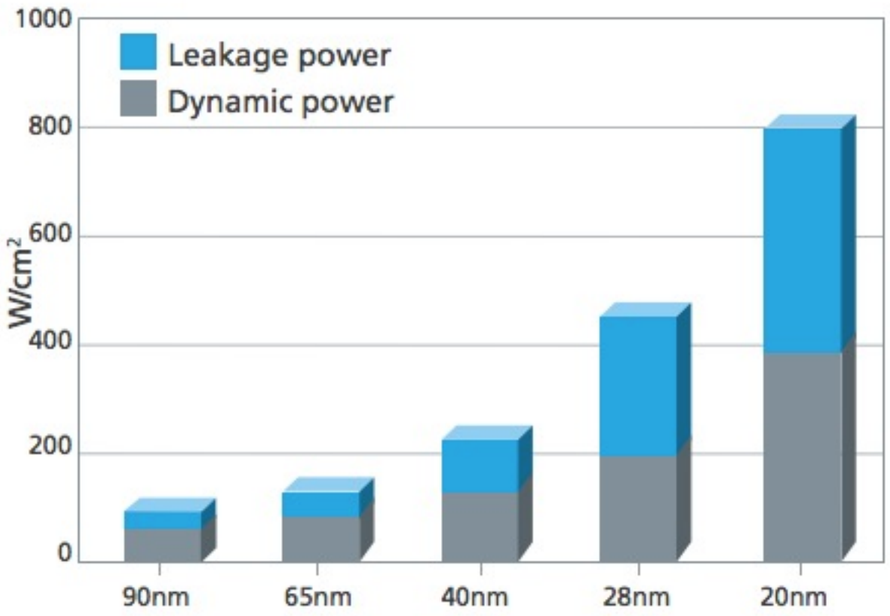


Figure 1: Leakage power becomes a growing problem as demands for more performance and functionality drive chipmakers to nanometer-scale process nodes (Source: IBS).

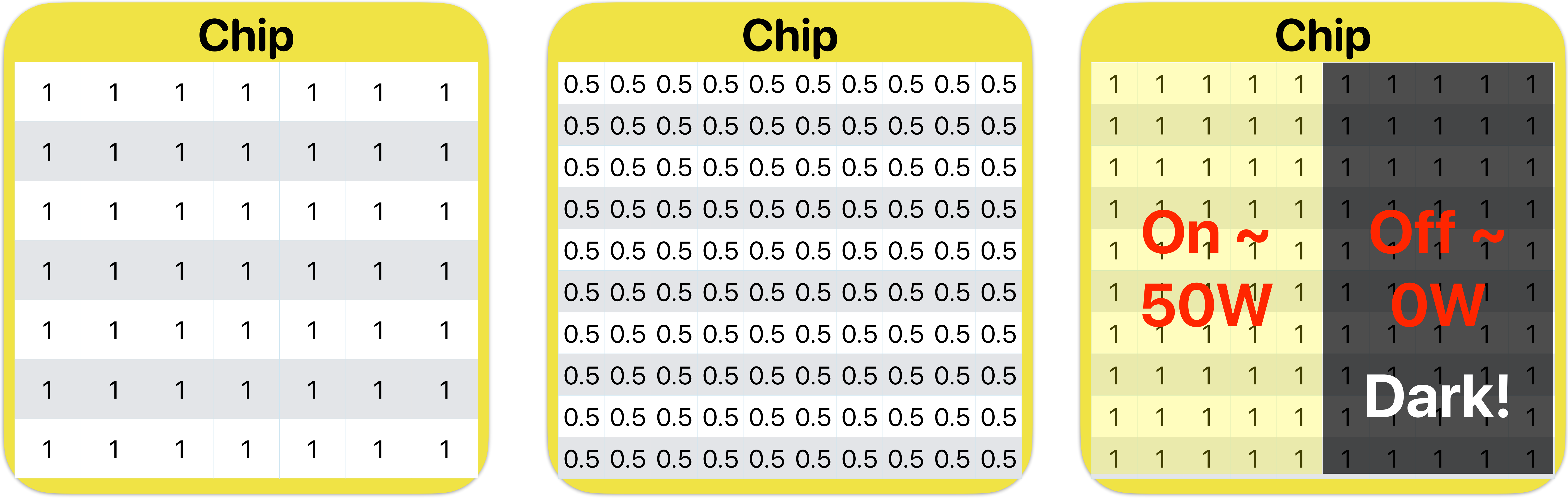
Dennardian Broken

- Given a scaling factor S

Parameter	Relation	Classical Scaling	Leakage Limited
Power Budget		1	1
Chip Size		1	1
Vdd (Supply Voltage)		1/S	1
Vt (Threshold Voltage)	1/S	1/S	1
tex (oxide thickness)		1/S	1/S
W, L (transistor dimensions)		1/S	1/S
Cgate (gate capacitance)	WL/tox	1/S	1/S
Isat (saturation current)	WVdd/tox	1/S	1
F (device frequency)	Isat/(CgateVdd)	S	S
D (Device/Area)	1/(WL)	S^2	S^2
p (device power)	IsatVdd	$1/S^2$	1
P (chip power)	Dp	1	S^2
U (utilization)	1/P	1	$1/S^2$

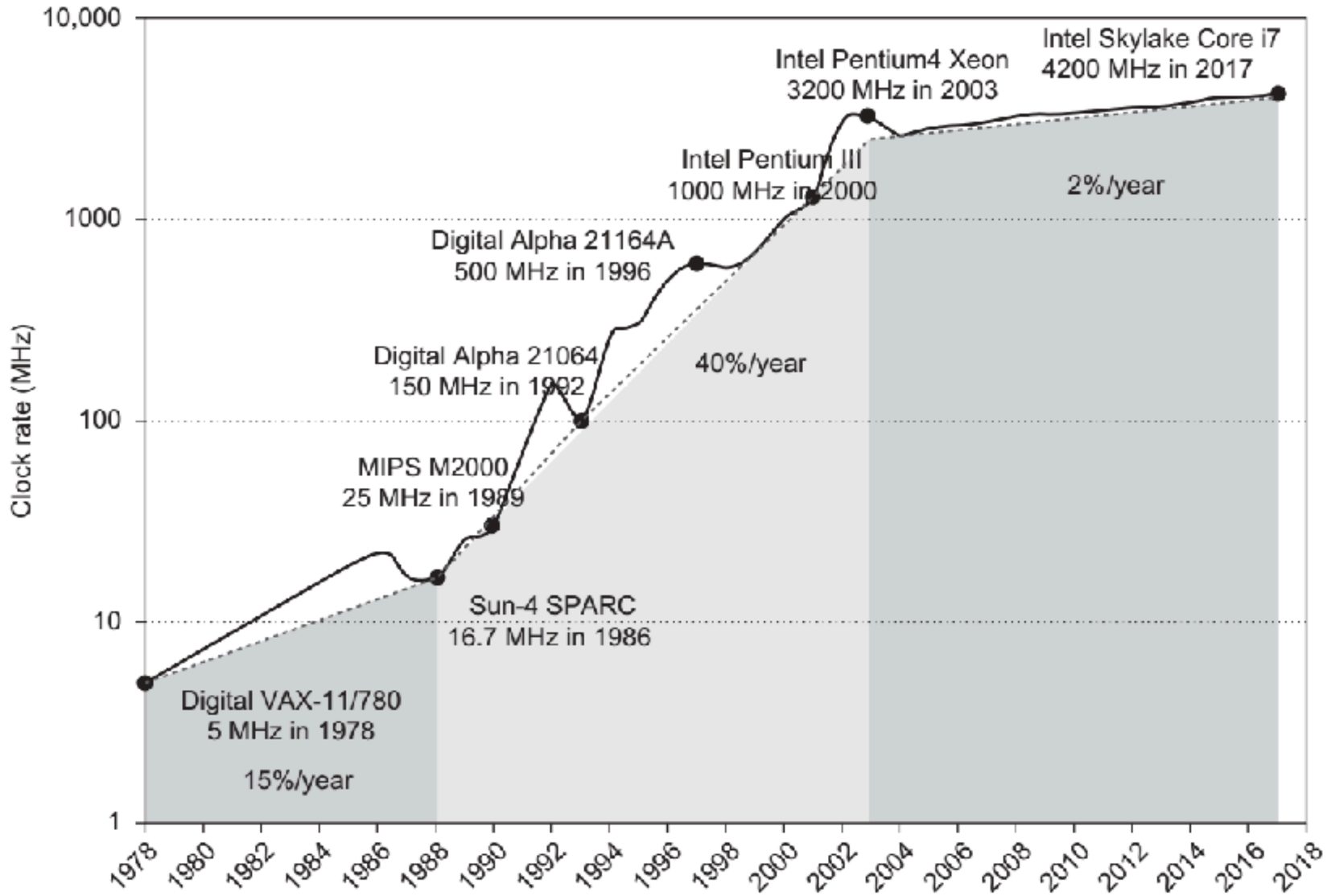
Power consumption to light on all transistors

Dennardian Scaling | Dennardian Broken



=49W | =50W | =100W!

Clock rate improvement is limited nowadays



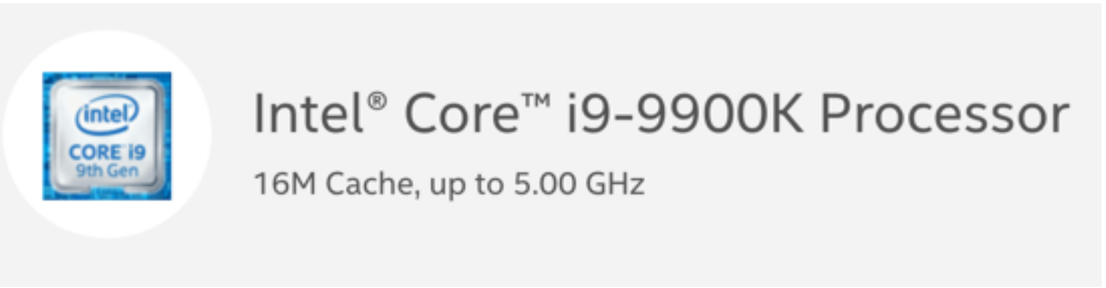
Solutions/trends in dark silicon era

Trends in the Dark Silicon Era

- Aggressive dynamic voltage/frequency scaling
- Throughout oriented — slower, but more
- Just let it dark — activate part of circuits, but not all
- From general-purpose to domain-specific — ASIC

Aggressive dynamic frequency scaling

Modern processor's frequency



Intel® Core™ i9-9900K Processor

16M Cache, up to 5.00 GHz

Essentials

Product Collection	
Code Name	
Vertical Segment	
Processor Number	
Status	
Launch Date	
Lithography	
Included Items	
Use Conditions	
Recommended Customer Price	

CPU Specifications

# of Cores	8
# of Threads	16
Processor Base Frequency	3.60 GHz
Max Turbo Frequency	5.00 GHz
Cache	16 MB Intel® Smart Cache
Bus Speed	8 GT/s
Intel® Turbo Boost Technology 2.0 Frequency‡	5.00 GHz
TDP	95 W

Dynamic/Active Power

- The power consumption due to the switching of transistor states
- Dynamic power per transistor

  $P_{dynamic} \sim \alpha \times C \times V^2 \times f \times N$

  - α: average switches per cycle
  - C: capacitance
  - V: voltage
  - f: frequency, usually linear with V
  - N: the number of transistors

Recap: Demo — changing the max frequency and performance

- Change the maximum frequency of the intel processor — you learned how to do this when we discuss programmer's impact on performance
- LIKWID a profiling tool providing power/energy information
  - likwid-perfctr -g ENERGY [command_line]
  - Let's try blockmm and popcount and see what's happening!

Slower, but more

More cores per chip, slower per core

Products Solutions Support

intel

	Intel® Xeon® Processor E7-8890 v4	Intel® Xeon® Processor E7-8893 v4	Intel® Xeon® Processor E7-8880 v4
Status	Launched	Launched	Launched
Launch Date	Q2'16	Q2'16	Q2'16
Lithography	14 nm	14 nm	14 nm
Performance			
# of Cores	24	4	22
# of Threads	48	8	44
Processor Base Frequency	2.20 GHz	3.20 GHz	2.20 GHz
Max Turbo Frequency	3.40 GHz	3.50 GHz	3.30 GHz
Cache	60 MB	60 MB	55 MB
Bus Speed	9.6 GT/s	9.6 GT/s	9.6 GT/s
# of QPI Links	3	3	3
TDP	165 W	140 W	150 W

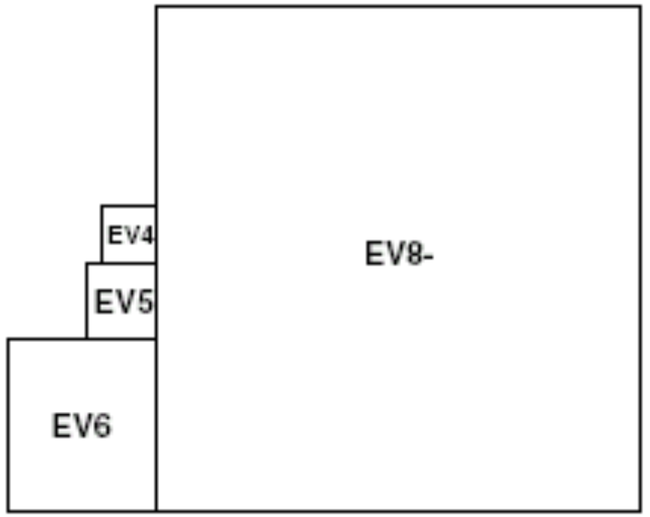
Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction

Rakesh Kumar, Keith Farkas, Norm P. Jouppi, Partha Ranganathan, Dean M. Tullsen.
University of California, San Diego and HP Labs

Areas of different processor generations

- You fit about 5 EV5 cores within the same area of an EV6
- If you build a quad-core EV6, you can use the same area to
  - build 20-core EV5
  - 3EV6+5EV5

Processor	EV5	EV6	EV6+
Issue-width	4	6 (OOO)	6 (OOO)
I-Cache	8KB, DM	64KB, 2-way	64KB, 2-way
D-Cache	8KB, DM	64KB, 2-way	64KB, 2-way
Branch Pred.	2K-gshare	hybrid 2-level	hybrid 2-level
Number of MSHRs	4	8	16
Number of threads	1	1	4
Area (in mm^2)	5.06	24.5	29.9



Poll close in 1:30

Single ISA heterogeneous CMP

- Regarding "Single-ISA Heterogeneous Multi-Core Architectures", how many of the following statements is/are correct?
  - ① You need to recompile and optimize the binary for each core architecture to exploit the thread-level parallelism in this architecture
  - ② For a program with limited thread-level parallelism, single ISA heterogeneous CMP would deliver better or at least the same level of performance than homogeneous CMP
  - ③ For a program with rich thread-level parallelism, single ISA heterogeneous CMP would deliver better or at least the same level of performance than homogeneous CMP built with older-generation cores
  - ④ Spending more instructions on older-generation cores would always lead to better energy-delay

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

Single ISA heterogeneous CMP

- Regarding "Single-ISA Heterogeneous Multi-Core Architectures", how many of the following statements is/are correct?
  - ① You need to recompile and optimize the binary for each core architecture to exploit the thread-level parallelism in this architecture
  - ② For a program with limited thread-level parallelism, single ISA heterogeneous CMP would deliver better or at least the same level of performance than homogeneous CMP
  - ③ For a program with rich thread-level parallelism, single ISA heterogeneous CMP would deliver better or at least the same level of performance than homogeneous CMP built with older-generation cores
  - ④ Spending more instructions on older-generation cores would always lead to better energy-delay

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Announcement

- Last lecture this Wednesday — we will take a group photo — prepare your virtual background!
- Project due TONIGHT — no extension
- Assignment #5 due Wednesday — the last assignment
- iEVAL until 12/11
  - Please fill the survey to let us know your opinion!
  - Don't forget to take a screenshot of your submission and submit through iLearn — it counts as a **full credit assignment**
  - **We will drop your lowest 2 assignment grades**
- Talk by Reetu Das today @ 11am
  - Attend and submit a screenshot, count as a full credit reading quiz
  - Read instruction through iLearn
- Office Hours on Zoom (the office hour link, not the lecture one)
  - Hung-Wei/Prof. Usagi: M 8p-9p, W **2p-4p — the last office hour by Prof. Usagi**
  - Quan Fan: F 1p-3p

Announcement — final exam

- Final Exam
  - The final can be opened only once -- if you accidentally close the browser or the browser crashes or your lose Internet connection, you cannot re-initiate it and we WILL NOT help you for these cases. Browsers crash and accidental closing of tabs occur a lot when you have many opened tabs. Please be careful.
  - Q21 - Q28 are comprehensive exam questions -- You must receive at least 60% from Q21-Q25 AND 60% from Q26-Q28 to be considered as PASS
  - This final covers EVERYTHING mentioned/assigned this quarter.
  - This is an open-book, open-note test, but again, the more you open, the higher chance your computer will have issues.
  - We have MANY questions for you, but you only have a total of 180 minutes to finish. Heavily rely on your notes/book/ cheatsheets is not a good idea.
  - Please show your work if appropriate -- we don't give credits to answers only have the final result
  - There is no partial credits for multiple choice questions. Please think thoroughly.
  - Reference online solution, discuss with ANY other human being or digital assistant (e.g, Siri, Google Home, Alexa or whatever you name it) is considered as cheating.
  - We will not automatically submit your test when time is up. If your submission is late by x sec, your grade is max(raw_score * ((100-x)/100),0)
  - Will release a sample final at the end of the last lecture

Computer
Science &
Engineering

203

つづく