Computer Abstractions and Technology

Prof. Rajib Ranjan Maiti
CSIS, BITS-Pilani, Hyderabad

Power and Energy

Clock and power

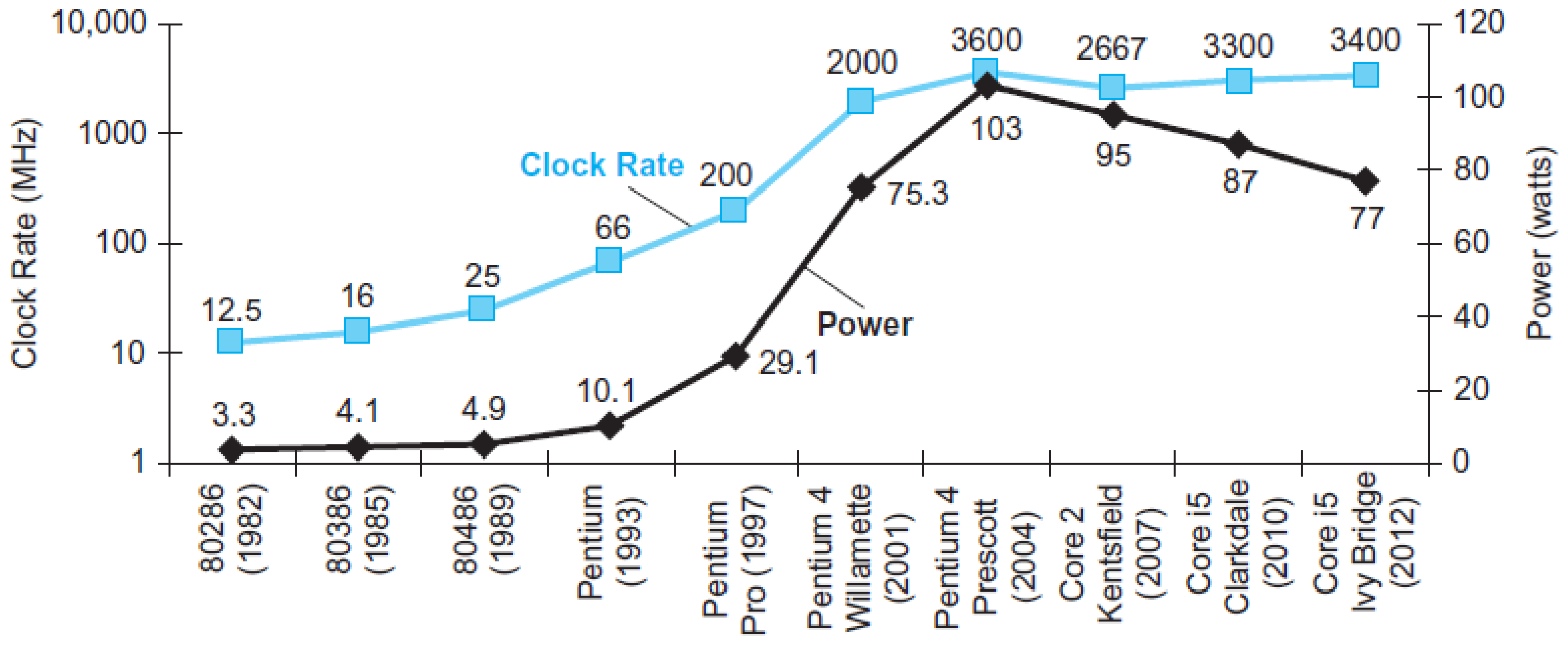


FIGURE 1.16 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

Clock and power

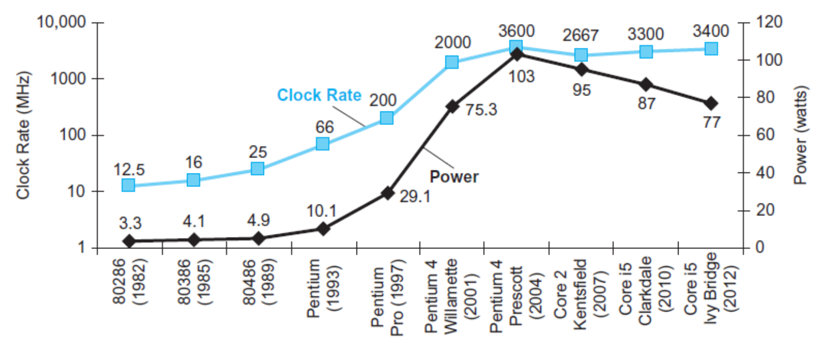


FIGURE 1.16 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

- Both clock rate and power increased rapidly for decades, and then flattened off
  - the practical **power** limit exhausted for cooling commodity microprocessors
  - In other words, power provides a limit to what we can cool
- The architects of warehouse scale computers try to reduce the costs of powering and cooling 100,000 servers
  - as the costs are high at this scale
- The energy metric joules
  - A better measure than power rate using watts
  - Basically, energy = power x time

Clock and power

- The dominant technology for IC is called CMOS
- In CMOS, the primary energy consumption is
  - dynamic energy, i.e., consumed when transistors switch states, e.g., 0 → 1 or 1 → 0

- Dynamic energy
  - depends on the capacitive loading of each transistor and the voltage applied

- *Dynamic Power equation*
  - $P_d \; \alpha \; \frac{1}{2} C_l V^2 F,$
    - P_d ≡ Power Dynamic, C_l ≡ Capacitive Load, V ≡ voltage, F ≡ Frequency switches

- *Static Power equation*
  - $P_s \propto Cr_s V,$
    - P_s ≡ Power Static, Cr_s ≡ Current Static, V ≡ voltage

Example

- Suppose we developed a new, simpler processor that has 85% of the capacitive load of the more complex older processor.
- Further, assume that it has adjustable voltage so that it can reduce voltage 15% compared to old processor, which results in a 15% shrink in frequency of switching.
- What is the impact on dynamic power?

Example

- Suppose we developed a new, simpler processor that has 85% of the capacitive load of the more complex older processor.
- Further, assume that it has adjustable voltage so that it can reduce voltage 15% compared to old processor, which results in a 15% shrink in frequency of switching.
- What is the impact on dynamic power?

- $P_d \; \alpha \; \frac{1}{2} C_l V^2 F,$
  - P_d ≡ Power Dynamic, C_l ≡ Capacitive Load, V ≡ voltage, F ≡ Frequency switched
- $P_s \propto Cr_s V,$
  - P_s ≡ Power Static, Cr_s ≡ Current Static, V ≡ voltage

$$\frac{\text{Power}_{\text{new}}}{\text{Power}_{\text{old}}} = \frac{\langle \text{Capacitive load} \times 0.85 \rangle \times \langle \text{Voltage} \times 0.85 \rangle^2 \times \langle \text{Frequency switched} \times 0.85 \rangle}{\text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency switched}}$$

Thus the power ratio is

$$0.85^4 = 0.52$$

Hence, the new processor uses about half the power of the old processor.

SPEC CPU Benchmark

- A computer user
  - would be the perfect candidate to evaluate a new computer
- To evaluate
  - Need: two computer systems
  - Run and Compare: the execution time of a workload on the two systems
- Alternative approach
  - use a set of benchmarks—programs specifically chosen to measure performance
- SPEC (System Performance Evaluation Cooperative)
  - An effort funded and supported by a number of computer vendors
  - Aim is to create standard sets of benchmarks for modern computer systems

SPEC CPU Benchmark

- The latest is SPEC CPU2006, consists of
  - a set of 12 integer benchmarks (CINT2006), and
  - A set of 17 floating-point benchmarks (CFP2006)

- The integer benchmarks includes
  - a part of a C compiler
  - a chess program
  - a quantum computer simulation

- The floating-point benchmarks include
  - structured grid codes for **finite element modeling**,
  - particle method codes for **molecular dynamics**, and
  - sparse linear algebra codes for **fluid dynamics**.

SPECINTC2006 on Intel Core i7

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

FIGURE 1.18 SPECINTC2006 benchmarks running on a 2.66 GHz Intel Core i7 920. As the equation on page 35 explains, execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECINTC2006 is the geometric mean of the SPECratios.

SPECratio:
a bigger numeric result
➔ faster performance

Fallacies and Pitfalls

Fallacies and Pitfalls

- *Fallacy: Computers have been built in the same, old-fashioned way for far too long, and this antiquated model of computation is running out of steam.*

- *Pitfall: Ignoring the inexorable progress of hardware when planning a new machine.*

Fallacy and Pitfall

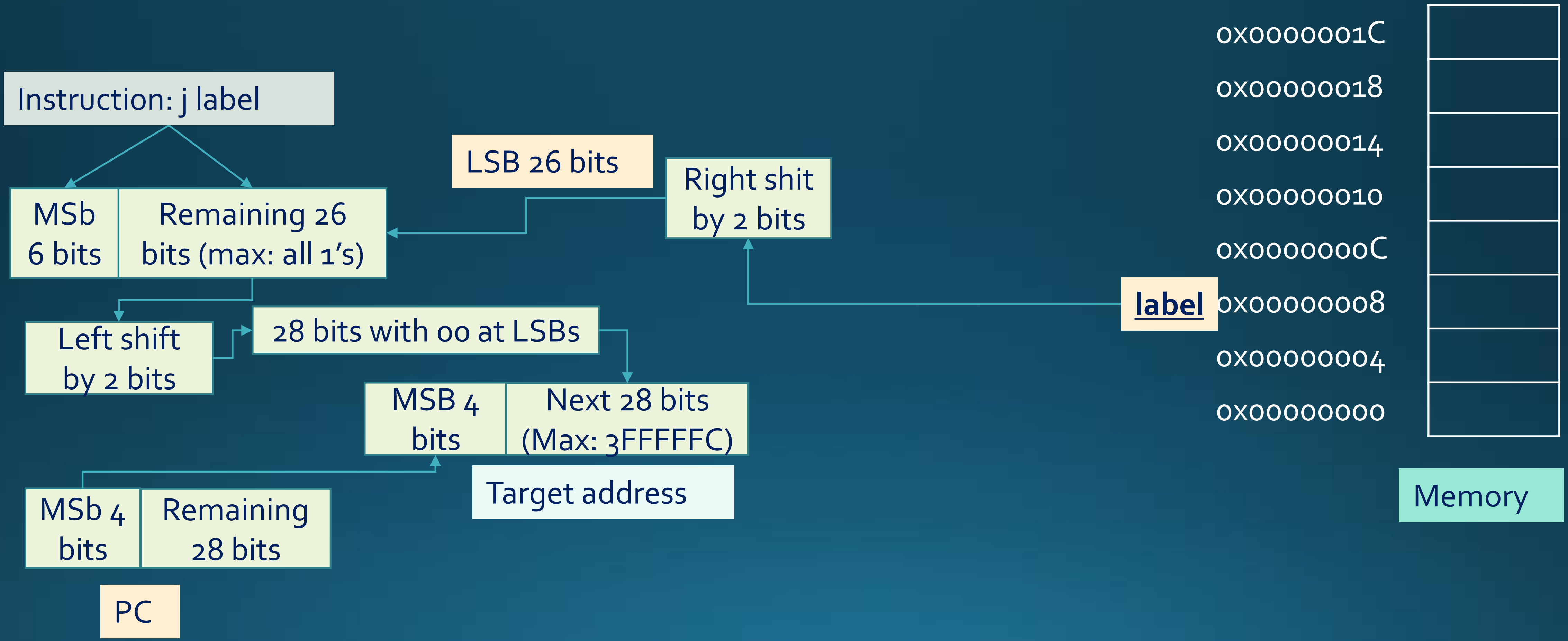
- *Pitfall:*
  - *Expecting the improvement of one aspect of a computer to increase overall performance by an amount proportional to the size of the improvement.*
  - *Example:*
    - Suppose a program runs in 100 seconds on a computer, with multiply operations responsible for 80 seconds of this time. How much do I have to improve the speed of multiplication if I want my program to run five times faster?
    - Ans.
      - Exec. time after improvement = exec. time unaffected + (Exec. Time affected / amount of improvement)
    - $100/5 = (100 – 80) + (80 / n) \Rightarrow 80/n = 0$????

Thank You

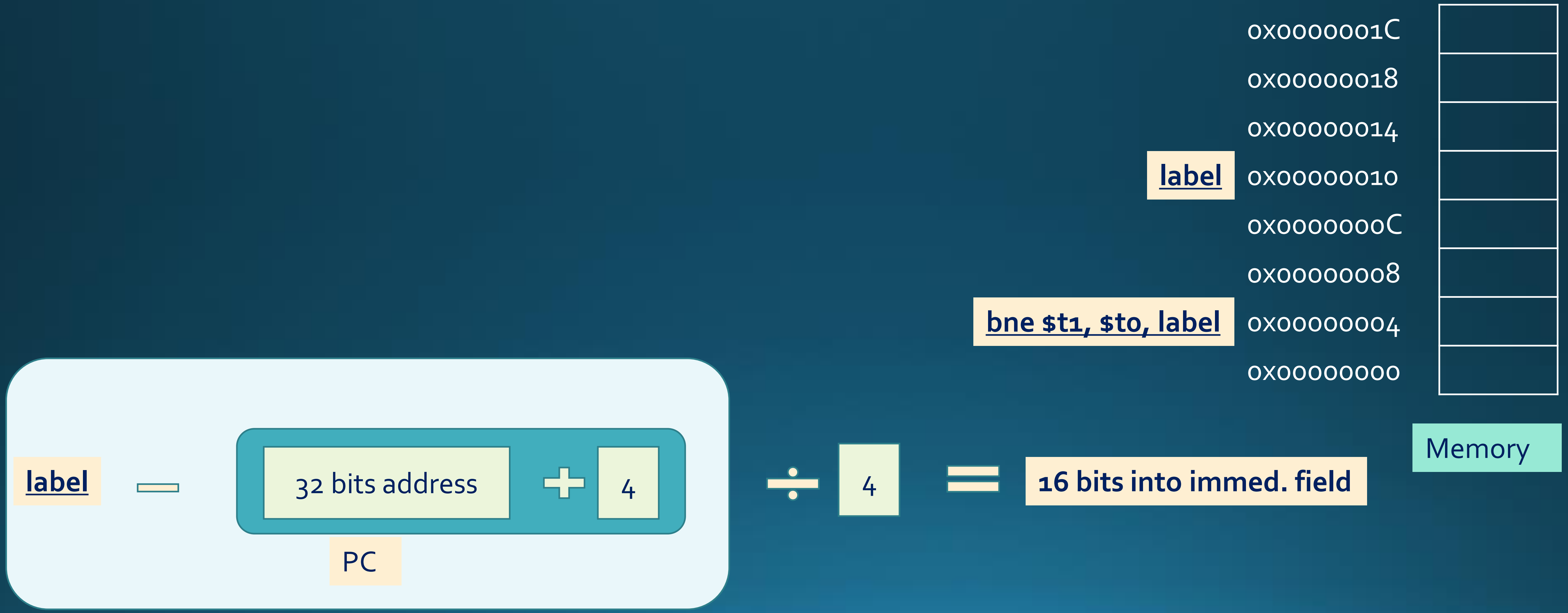
Backup

Address calculation : j label

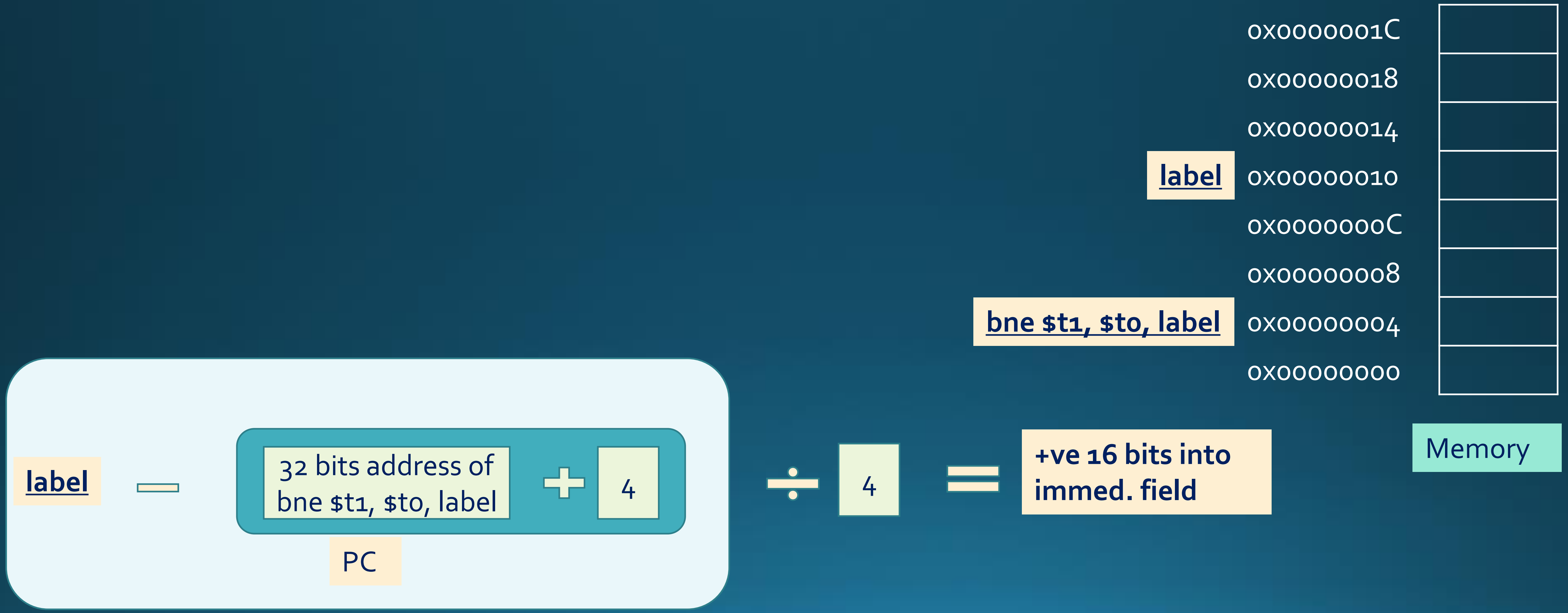
Effective address calculation for **j label** instruction:



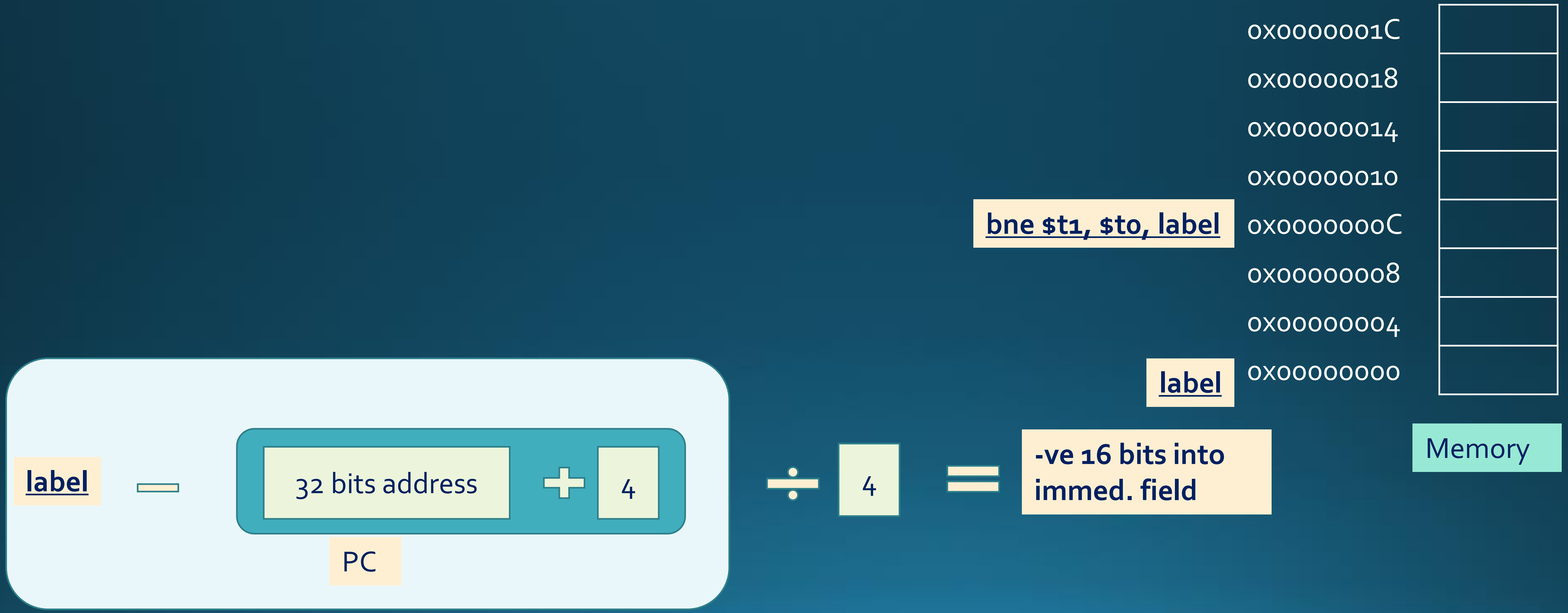
Address calculation: bne $t1, $to, label



Address calculation: bne $t1, $t0, label



Address calculation: bne $t1, $t0, label



Address calculation: bne $t1, $t0, label

- Maximum +ve displacement using branch:
  - Highest +ve number in 16 bits = $(0111\ 1111\ 1111\ 1111)_2$
  - In 16 bits ⟶ $+2^{15} - 1 = +32{,}767$ (words)
  - Highest (+ve) Offset × 4 = 32,767 * 4 = +131,068 Bytes = +128Kibibytes
    - $(0111\ 1111\ 1111\ 1111)_2 = (01\ 1111\ 1111\ 1111\ 1100)_2$ = 0x1FFFC
  - Least -ve number in 16 bits = $(1000\ 0000\ 0000\ 0000)_2$
  - In 16 bits ⟶ $-2^{15} = -32{,}768$ words
  - Least (-ve) Offset × 4 = −32,768 * 4 = −131,072 Bytes = -128Kibibytes
    - (1000 0000 0000 0000)2 = (10 0000 0000 0000 0000)2 = 0x20000
- Range in words = - 32,768 to 32,767
- Range in bytes = ± 128KB
- Range in Hex = + 0x1FFFC to – 0x20000