Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and **Valerio Selis**

dmc@liverpool.ac.uk
V.Selis@liverpool.ac.uk

UNIVERSITY OF LIVERPOOL

Outline

- Recap
- Memory
- Types of silicon memory
  - Types of ROM
  - Types of RAM
- Speed VS Area
- Cache memory

Recap – Shift

Bit patterns are rotated by using the **barrel shifter**

Shift by an immediate:

- LS**L**S rd, rm, **#imm** LS**R**S rd, rm, **#imm**
- AS**R**S rd, rm, **#imm**

Where: - **#imm** is of 5 bits (0 ≤ **imm** ≤ 31)
- rd and rm must be low registers

Shift by a register:

- LS**L**S **ry**, **ry**, **rz** LS**R**S **ry**, **ry**, **rz**
- AS**R**S **ry**, **ry**, **rz** RO**R**S **ry**, **ry**, **rz**

Where: - **ry** and **rz** must be low registers
- Least significant byte of rz is used for the shift
- Source and destination registers must be the same

Recap – Subroutine

Group of instructions which are repeated many times

From main program to subroutine:

- BL <target_addr> BL <label>
  - **32 bits** machine code
  - 24 bits for the target address (±16 MiB)
  - Value of the pc (r15) copied into the lr (r14)
- BLX rz
  - Branch to any point in the memory
  - Value of rz copied into the pc (r15)
  - Return address copied into the lr (r14)

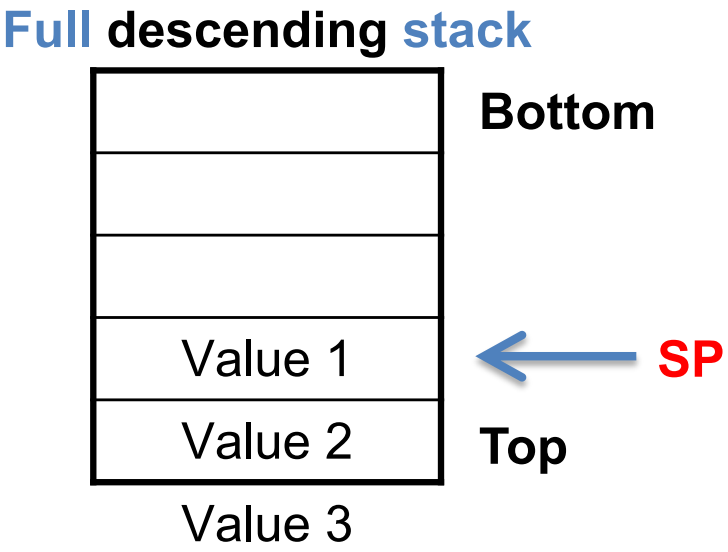
From subroutine to main program:

- BX rz BX lr BX r14
  - Value of rz or lr (r14) copied into the pc (r15)
  - Branch to any point in the memory

Recap – Stack

Area of computer memory identified by the stack pointer (SP – r13)

- It is a **L**ast **I**n **F**irst **O**ut (**LIFO**) queue
- Used for nested subroutines
- PUSH {<list registers>, lr}
  - Add values into the stack
- POP {<list registers>, pc}
  - Remove values from the stack



Recap – Pipeline

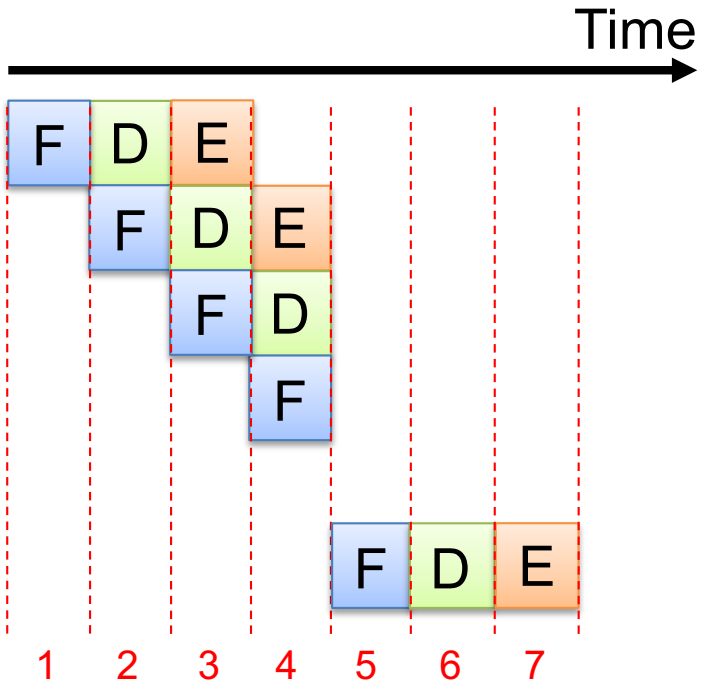
Feature that allows to execute simultaneously *n* instructions in an *n* stage pipeline

- ARM Cortex M0: three stage pipeline (fetch, decode and execute)
- Performance of a program is obtained by using clock cycle per instruction (CPI):
  - Number of clock cycles divided by instructions executed
  - Best performance when CPI = 1
  - Unconditional branch, load and store: 2 clock cycles
  - Conditional branch: 2 clock cycles only when executed

Recap – Pipeline and CPI example

In the ARM Cortex M0, the CPU can simultaneously **execute** an instruction, **decode** the next instruction and **fetch** the next but one instruction.

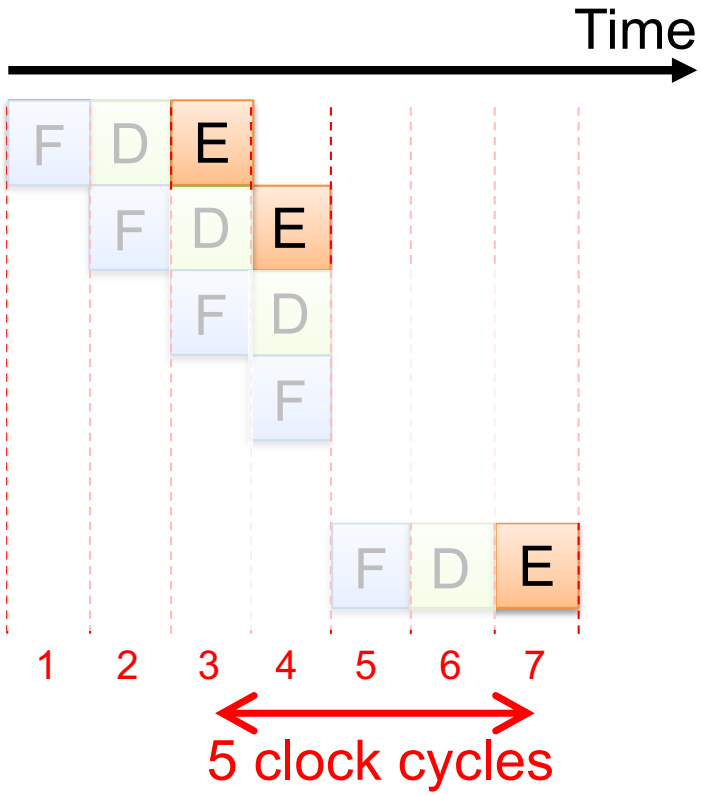
	Instruction	Address
	MOVS r0, #0	0x00008000
	BEQ jump	0x00008002
	SUBS r1, r2, r2	0x00008004
	ORRS r2, r4	0x00008006
	ANDS r3, r5	0x00008008
jump	ADDS r6, r0, #1	0x0000800A



Recap – Pipeline and CPI example

In the ARM Cortex M0, the CPU can simultaneously **execute** an instruction, **decode** the next instruction and **fetch** the next but one instruction.

	Instruction	Address
	MOVS r0, #0	0x00008000
	BEQ jump	0x00008002
	SUBS r1, r2, r2	0x00008004
	ORRS r2, r4	0x00008006
	ANDS r3, r5	0x00008008
jump	ADDS r6, r0, #1	0x0000800A



5/**3** = 1.66 CPI

UNIVERSITY OF LIVERPOOL

Memory

Memory can be defined as any device that can store information.

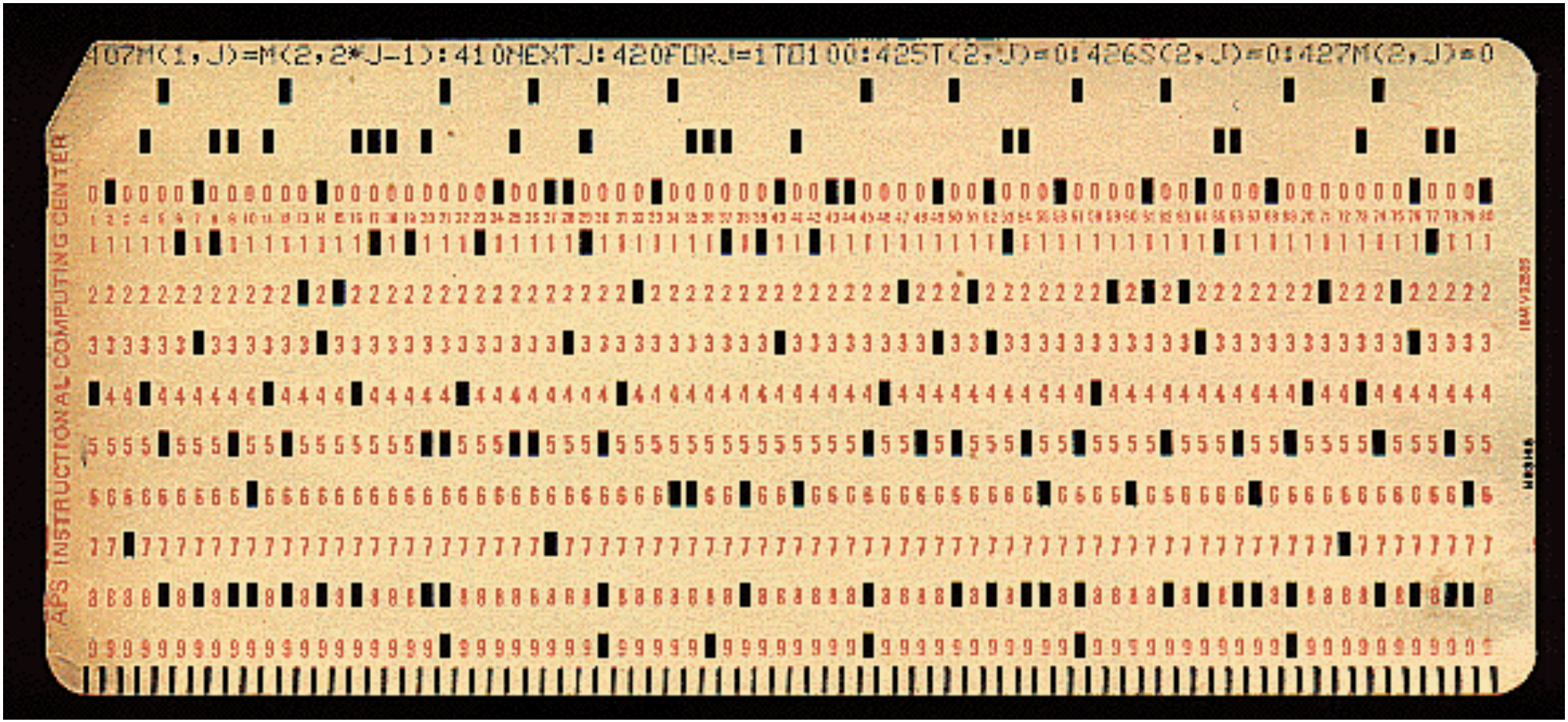
In the context of microprocessors, memory is a device that can store information in a **binary format**.

Historically computer memory included:

- punch cards, magnetic tape, magnetic core - all now obsolete.

Most memory is now in the form of semiconductor integrated circuits, hard disks, USB flash drive, etc.

Old memory technology – punch card



Used until mid-1970s

UNIVERSITY OF LIVERPOOL

512kBytes: IBM computer in the 1970's

UNIVERSITY OF LIVERPOOL

Old memory technology – floppy disk

In 2007 only 2% of PCs were sold with a floppy disk drive

8-inch	5¼-inch	3½-inch
568.320 kB	1,200 KB	1,440 KB

UNIVERSITY OF LIVERPOOL

Types of Silicon Memory

Silicon IC memory can be classified as either read-only or read-write.

Read-only memory (or ROM) is used for the operating system on a desktop computer or an application program in an embedded system such as a mobile phone.

Read-write memory (which is normally called RAM) is used for an application program on a desktop computer and for temporary data.

Types of ROM

There are several generations of silicon ROM.

Programmable ROM (PROM) could be electrically programmed once only and then the contents were fixed.

Erasable PROM (EPROM) could also have its contents erased by exposure to UV light and could then be reprogrammed.

Electrically erasable PROM (EEPROM) could have its contents erased by an electrical signal.

Types of ROM

Flash Erasable PROM (FEPROM or flash memory) is similar to EEPROM but erasing could only be done over a significant part of (maybe all) the integrated circuit.

All **ROM is non-volatile** - meaning the information stored in ROM memory is not lost when it is disconnected from a power source.

The data stored in ROM will remain almost indefinitely e.g. many tens of years.

RAM

In contrast to ROM, the contents of RAM memory will be lost almost as soon as power is switched off - **RAM is volatile**.

The acronym RAM stands for Random Access Memory

- The time taken to read or write data from or to the memory (the 'access time') does not depend upon the order in which the data is accessed.
- The memory locations can be accessed randomly or sequentially in the same time. This is not true for hard disks for example.

There are two types of RAM: Static and Dynamic RAM

Dynamic RAM

The contents of dynamic RAM (DRAM) must be refreshed every few milliseconds.

The binary data is held as an electrical charge on a tiny capacitor:

- A charged capacitor represents a binary “1”
- A discharged capacitor represents a binary “0”.

However the charge on the capacitor can leak away so it must be topped up or refreshed every few milliseconds.

Static RAM

Static RAM (SRAM) uses D type latches to store data and it does not need to be refreshed.

The data stored in a SRAM memory will be retained as long as it is connected to a power supply.

SRAM is generally faster than DRAM but needs a greater surface area of silicon to store the same amount of data:

- SRAM is more expensive than DRAM for the same amount of storage.

Memory: fast or big?

In general a bigger memory is a slower memory e.g. a hard disk can store terabytes but has an access time of tens of milliseconds.

The **main memory** of a microprocessor is typically DRAM with an access time of around 100 nanoseconds and DRAM chips hold up to 16 gigabytes of data.

Memory can be placed on the same IC as the processor (**'on chip' memory**) and may have an access time of 10 nanoseconds but only hold up to 32 kilobytes of data (SRAM).

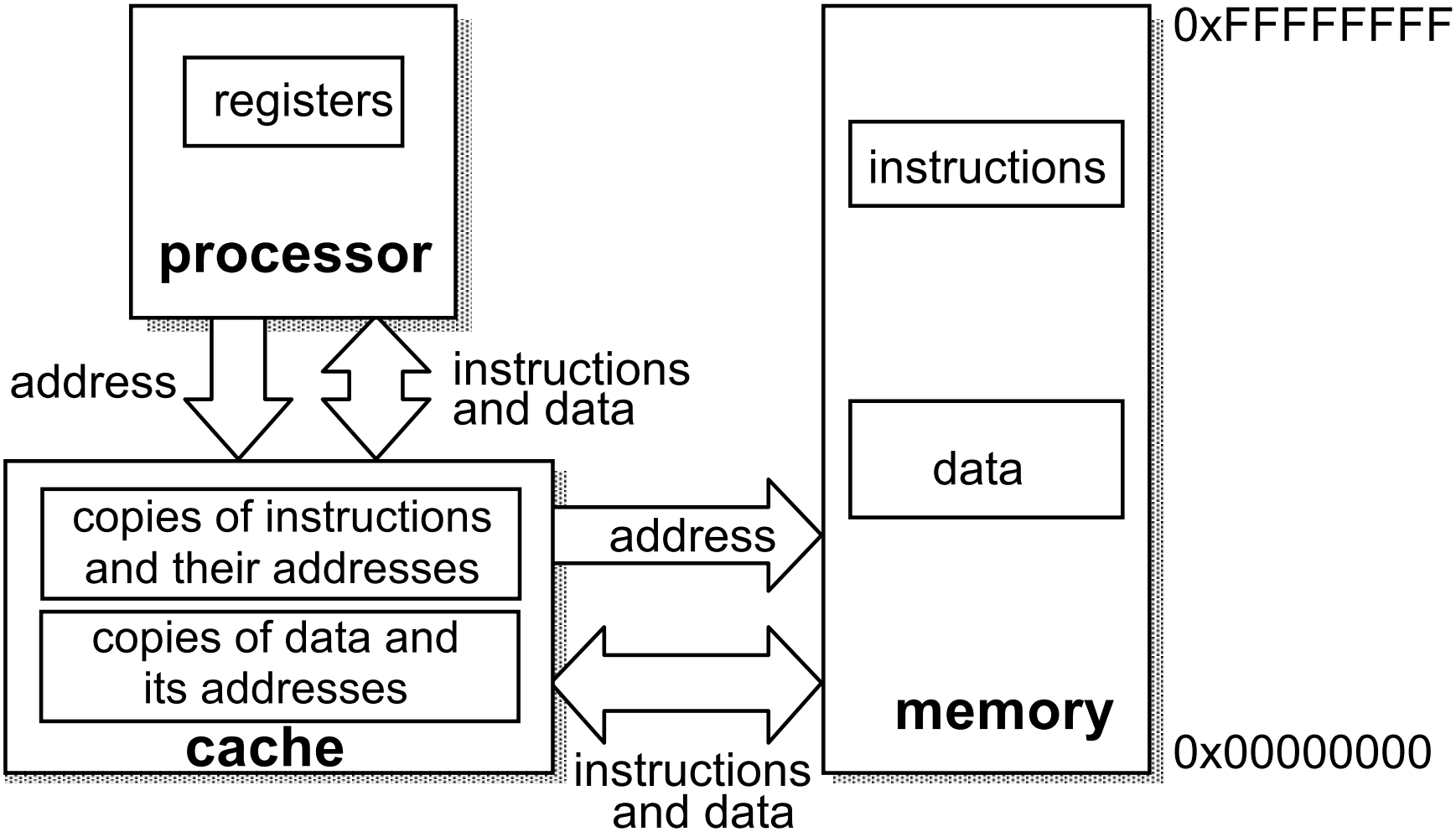
Fast and big!

However the microprocessor may be executing one instruction every 5 nanoseconds so how can a processor be connected to a very large memory with a very fast access time?

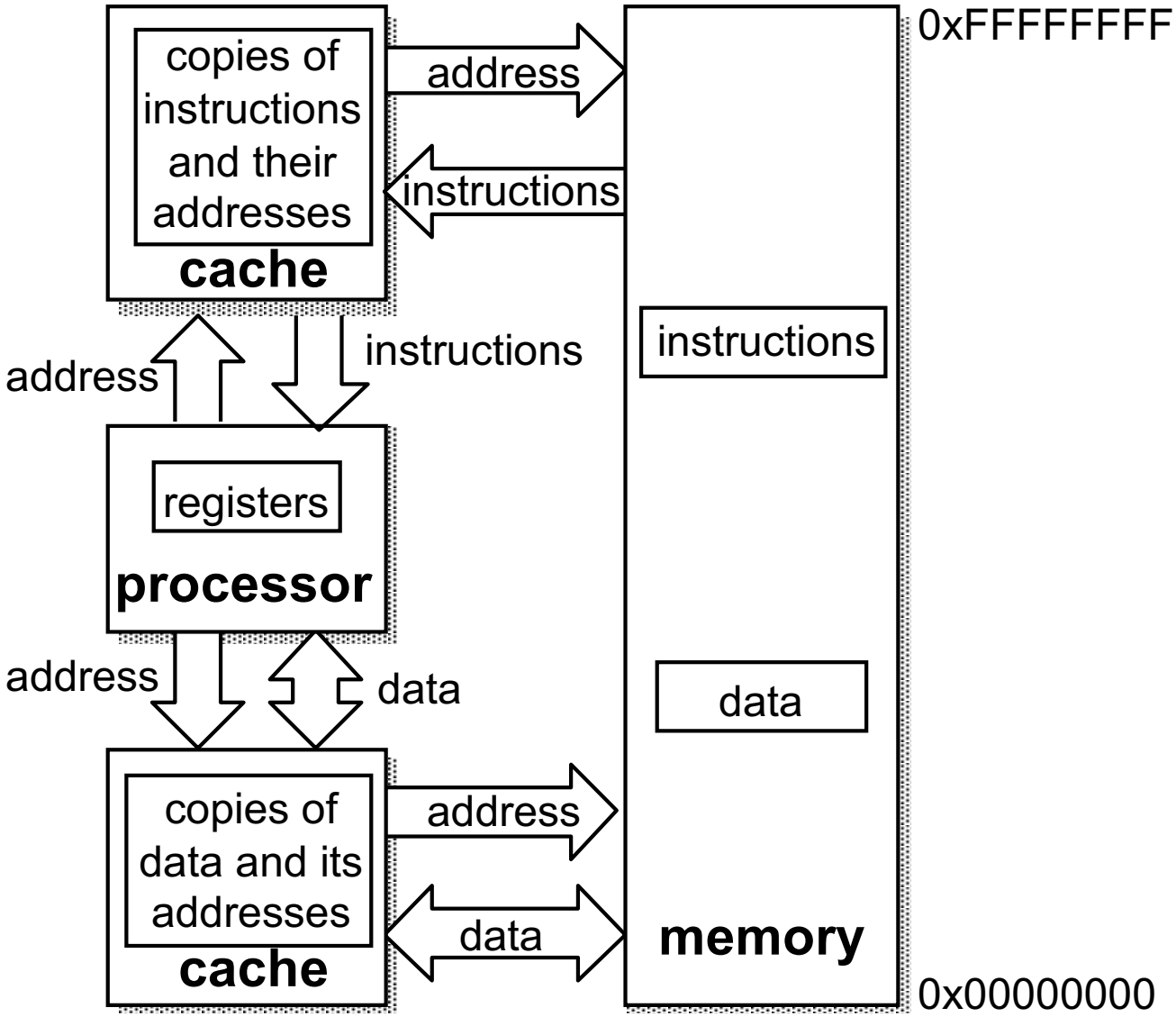
The key is to use a **memory cache.**

This is a small fast on chip memory that holds the most recently accessed data from the main memory.

A unified instruction and data cache



Separate data and instruction caches



This is often called the **modified Harvard architecture**.

Cache memory

When data is read from, or written to, main memory a copy of the data is saved in the memory cache (along with the main memory address)

When a subsequent read occurs the cache can be checked to see if the required data is already there:

- If it is (a cache ‘hit’) then it can be supplied immediately
- If not (a cache ‘miss’) then it must be fetched from main memory

Cache memory

Cache relies on two features of microprocessor programs:

- Temporal and spatial locality.

Temporal locality occurs because data accessed once is likely to be accessed again soon e.g. an instruction in a program loop.

Spatial locality occurs because data from one memory location is likely to be accessed if data in an adjacent memory location has recently been accessed. (Typically the data stored in the cache may be 16 adjacent bytes.)

Cache performance

The performance of a cache is measured by the **‘hit rate’** - that is the fraction of memory accesses satisfied by the cache.

The ‘hit rate’ should be over 90% to achieve the best results with modern microprocessors.

Sometimes there may be two levels of cache - a ‘**primary cache**’ that is on chip and an off chip ‘**secondary cache**’.

Also there may be separate caches for instructions and data - the ‘modified Harvard’ architecture.

Cache operation / behaviour

The ratio of number of hits to total accesses is the **HIT RATIO**, h.

The ratio of number of misses to total accesses is the **MISS RATIO**, m.

$$h = (1 - m) \quad \text{and} \quad m = (1 - h)$$

HIT ratios over 0.95:1 (95% hits) can be obtained by good cache design.

h will depend upon the program running.

Mean access time

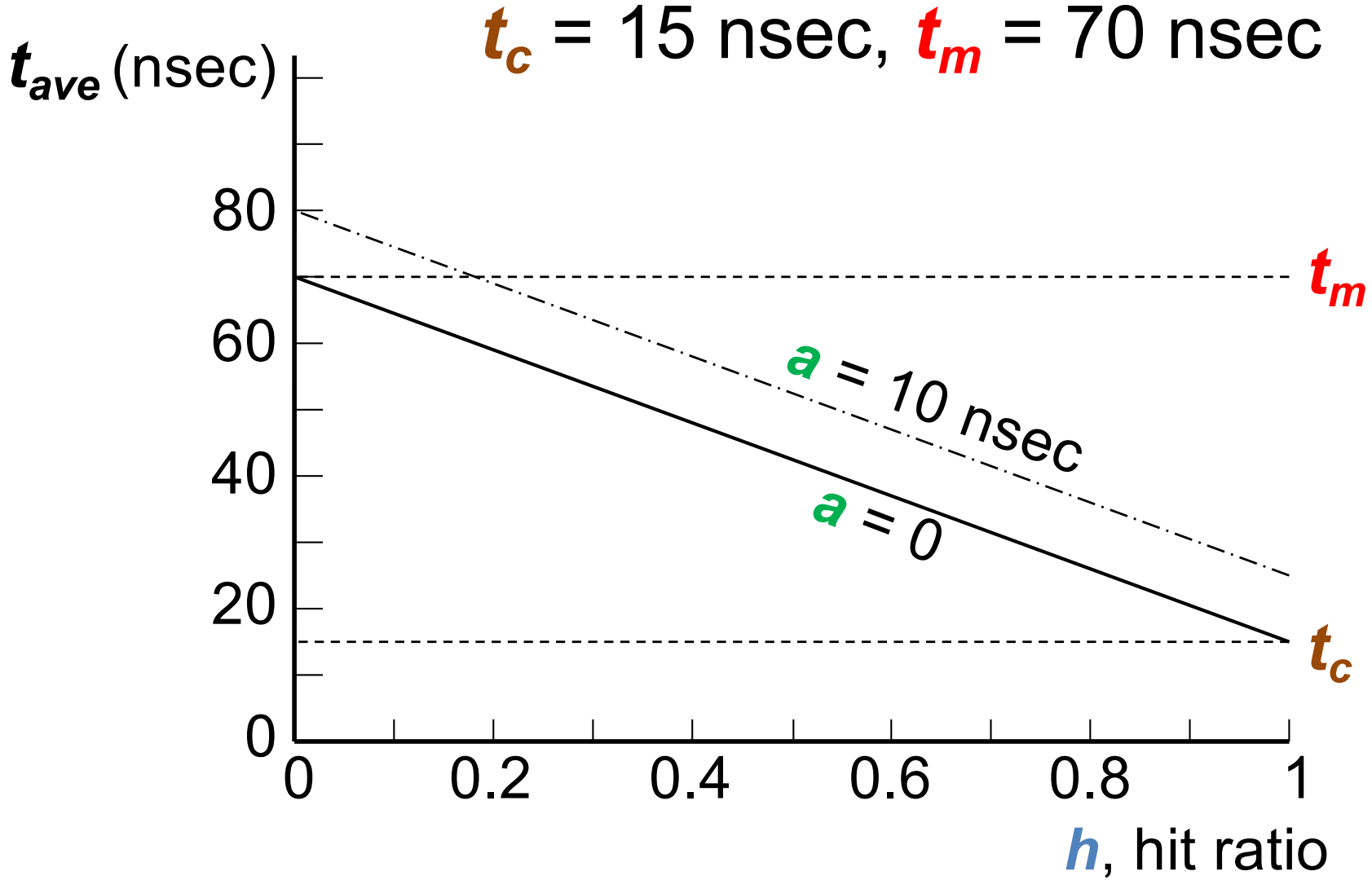
Simple approach (ignoring cache controller overheads) analysis suggests for many accesses with a cache access time of t_c, and a main memory access time of t_m the **mean access time** will be

$$t_{ave} = h \times t_c + m \times t_m = h \times t_c + (1 - h) \times t_m$$

In practice the cache control and routing circuits will add an extra time delay of a.

$$t_{ave} = h \times t_c + m \times t_m + a = h \times t_c + (1 - h) \times t_m + a$$

Mean access time



UNIVERSITY OF LIVERPOOL

Question

When poll is active, respond at **PollEv.com/elec211**

Text **ELEC211** to **22333** once to join

What is the average memory access time when
- the hit ratio is 0.95,
- the cache access time is 10 nsec,
- the main memory access time is 80 nsec and
- the extra time delay due to the cache control and routing circuits is 12 nsec?

- 24.5 nsec
- 25.5 nsec
- 26.5 nsec
- 27.5 nsec

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

Total Results

UNIVERSITY OF LIVERPOOL

Multiple levels of cache

Very complex multiple cache systems are used with high performance CPUs, these have multiple caches operating at different levels.

Size/performance of memory for small to medium size CPU systems such as ARM.

	Typical Size	Access time
CPU registers	64 to 256 bytes	1 cycle
Level 1 cache	16k to 64k bytes	1 to 2 cycles
Level 2 cache	128k to 1M bytes	8 cycles
Main memory	256M to Gbytes	16 cycles
Back-up	Very large	Block transfer

Summary

- Memory
- ROM and RAM
- Speed VS Area
- Cache memory

Next class?

Tomorrow at 2 p.m. in the
Building 502,
Lecture Theatre 2
(502-LT2)

Open book exam simulation (MUP)