MCU Interfacing

ENCE361: Design & Architecture: Lecture Block 2

Roadmap

		ENCE361 Lecture, Tutorial & Lab Schedule – 2020				v. 20.3		Updated 09/05/2020
		Lec 1		Lec 2		Lec 3		Lab
Wk	**Starting**	**Mon 1p**		**Wed 2p**		**Fri 2p**		**Mon 11a, Tue 9a, Tue 11a, Wed 11a**
10	**11 May**	23	Profiling	26	Load Analysis	27	MCU Interfacing	
11	**18 May**	28	Memory structures	29	MCU memory types	30	Arm Arithm./ Logic ccts	**Project demos in usual lab slot**
12	**25 May**	31	The ARM ISA	32	ARM Assembly language	33	Revision & Exam Prep	**Project report & code due Fri 29 May**

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Outline

- Hardware Interfacing
  - Input
  - Output

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Outline

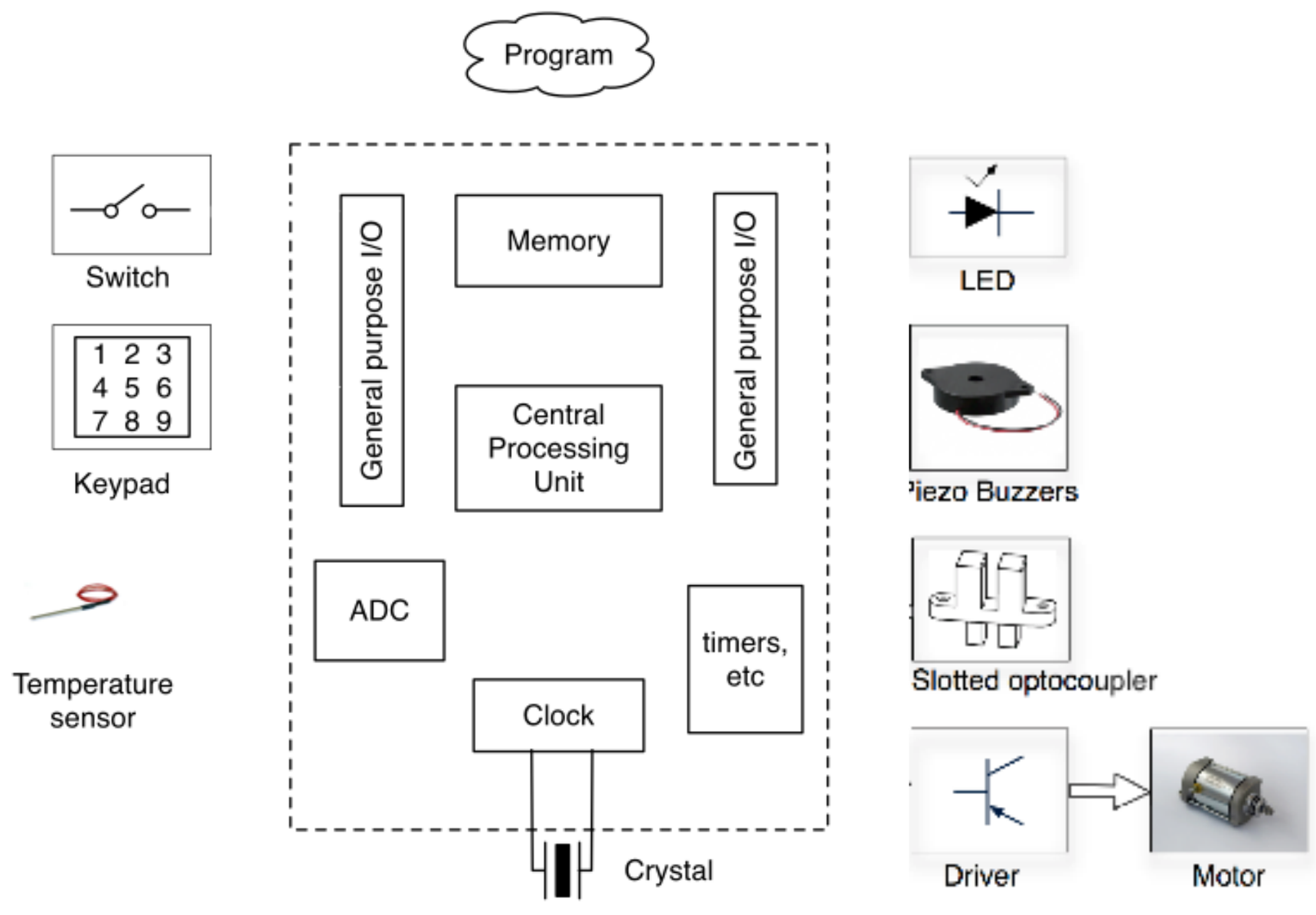
- Hardware Interfacing

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Hardware Interfacing

- An embedded application typically comprises:

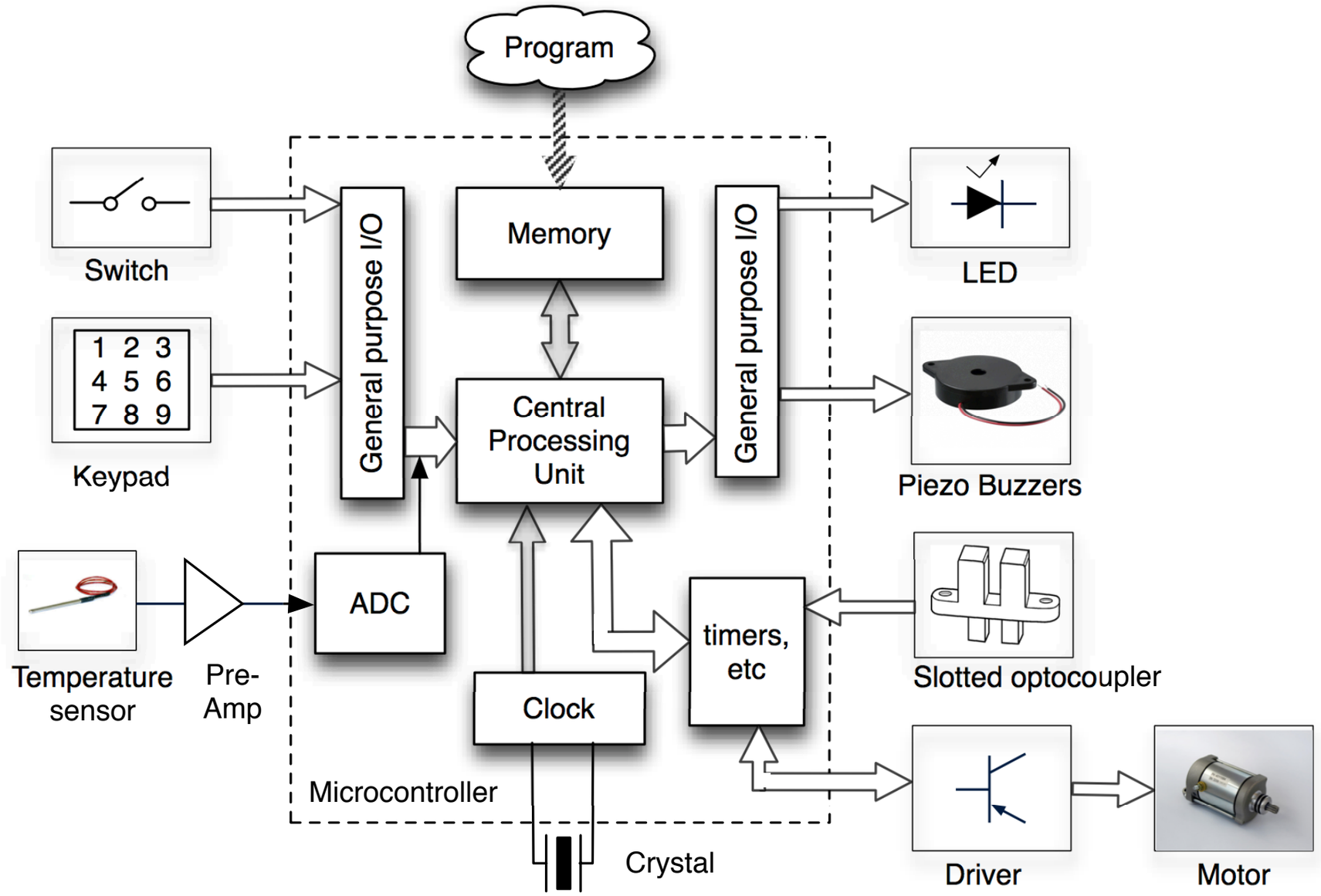
Sensors Microcontroller Actuators



Hardware Interfacing

- An embedded application typically comprises:

Sensors Microcontroller Actuators



Hardware Interfacing

- We're interested in how the separate components in embedded systems connect together electrically.
- This means:
  - Matching the operating voltages of the different components.
  - Ensuring that a component can source or sink enough current.
  - Assigning MCU resources judiciously to ensure all peripherals & I/O can be connected.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Microcontroller

- How many pins (GPIO, PWM, Analogue, I²C etc.) do we have to connect to peripherals?
- What is the operating voltage of the microcontroller?
    - Voltage for I/O may be different (higher) than core operating voltage.
- How much current can each MCU I/O pin source and sink?
    - Current capacity is not necessarily symmetric.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Microcontroller

- e.g. ATmega32/L

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA and 400.0mA TQFP/MLF

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Microcontroller

- e.g. TM4C123GH6PM

Table 24-7. GPIO Current Restrictions[a]

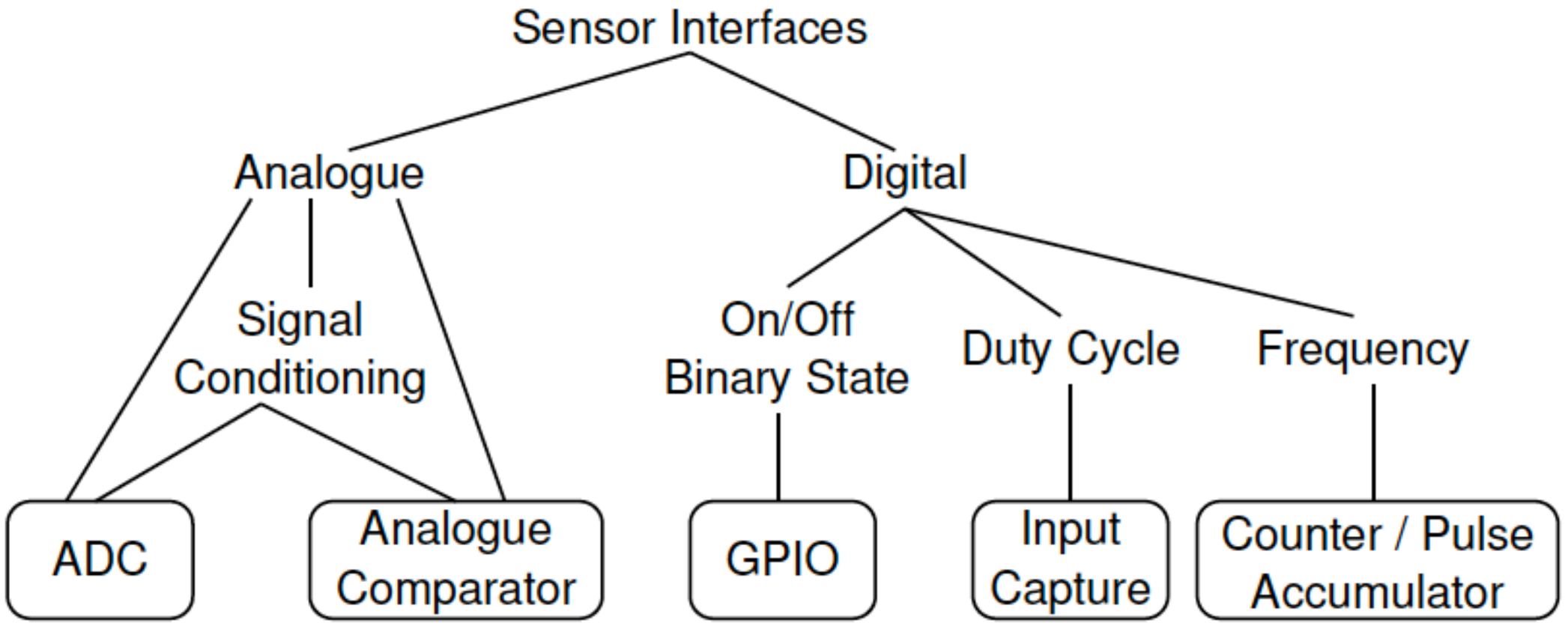
Parameter	Parameter Name	Min	Nom	Max	Unit
I_{MAXL}	Cumulative maximum GPIO current per side, left[b]	-	-	30	mA
I_{MAXB}	Cumulative maximum GPIO current per side, bottom[b]	-	-	35	mA
I_{MAXR}	Cumulative maximum GPIO current per side, right[b]	-	-	40	mA
I_{MAXT}	Cumulative maximum GPIO current per side, top[b]	-	-	40	mA

Table 24-8. GPIO Package Side Assignments

Side	GPIOs
Left	PB[6-7], PC[4-7], PD7, PE[0-3], PF4
Bottom	PA[0-7], PF[0-3]
Right	PB[0-3], PD[4-5]
Top	PB[4-5], PC[0-3], PD[0-3,6], PE[4-5]

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Inputs



Ciaran Moore
ciaran.moore@canterbury.ac.nz

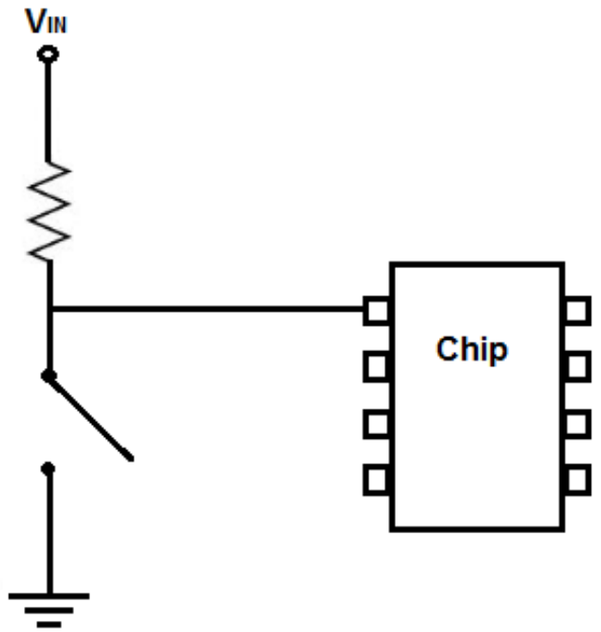
Inputs

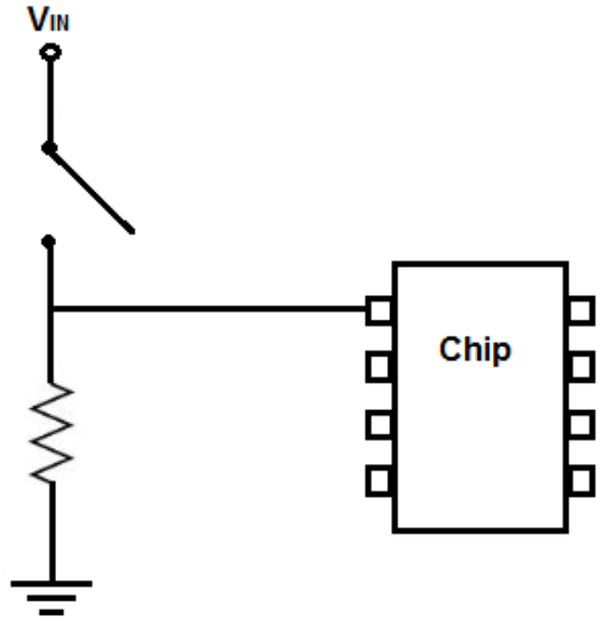
- Buttons & Switches
  - Pull-up or pull-down resistor
  - Debouncing
- Keypads
  - Pin multiplexing

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Buttons & Switches

- Pull-up or pull-down resistor
  - When a switch is closed the voltage on the MCU pin is forced to the voltage rail (V_{CC} or GND).
  - What is the voltage on the MCU pin when the switch is open?
- Unless they are connected to a rail, open switches look like antenna, whose voltage will change as a result of electromagnetic interference (EMI).
- Use a pull-up (or -down) resistor to force the pin voltage to a rail when the switch is open.

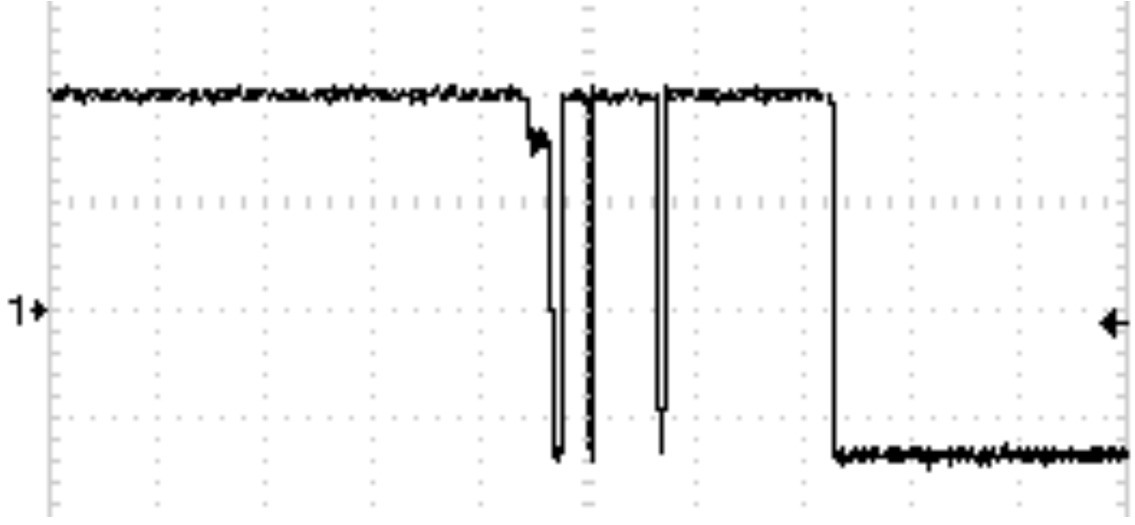


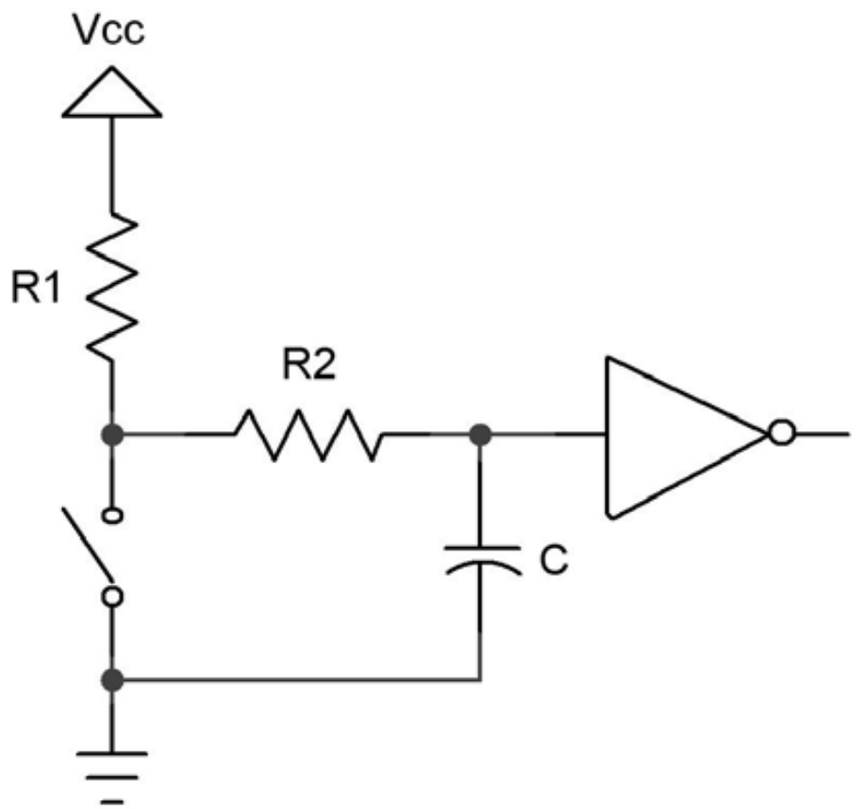


Ciaran Moore
ciaran.moore@canterbury.ac.nz

Buttons & Switches

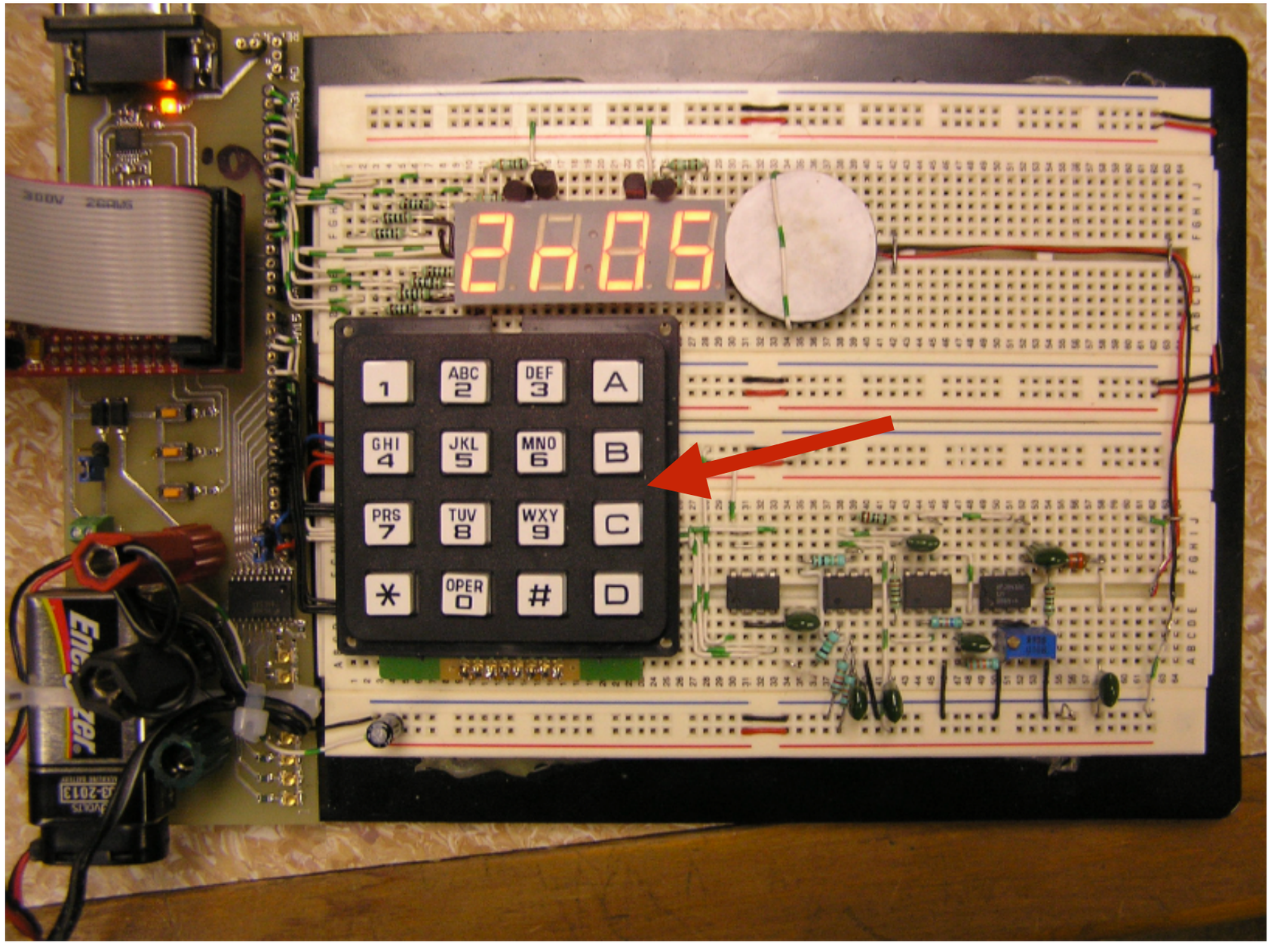
- Debouncing
  - Mechanical switches experience vibration during a change; this vibration manifests as multiple open/close events (typ. ≤ 10 ms* timescale).
  - Can "debounce" in hardware (RC network) or software (shift register, delay).





Ciaran Moore
ciaran.moore@canterbury.ac.nz

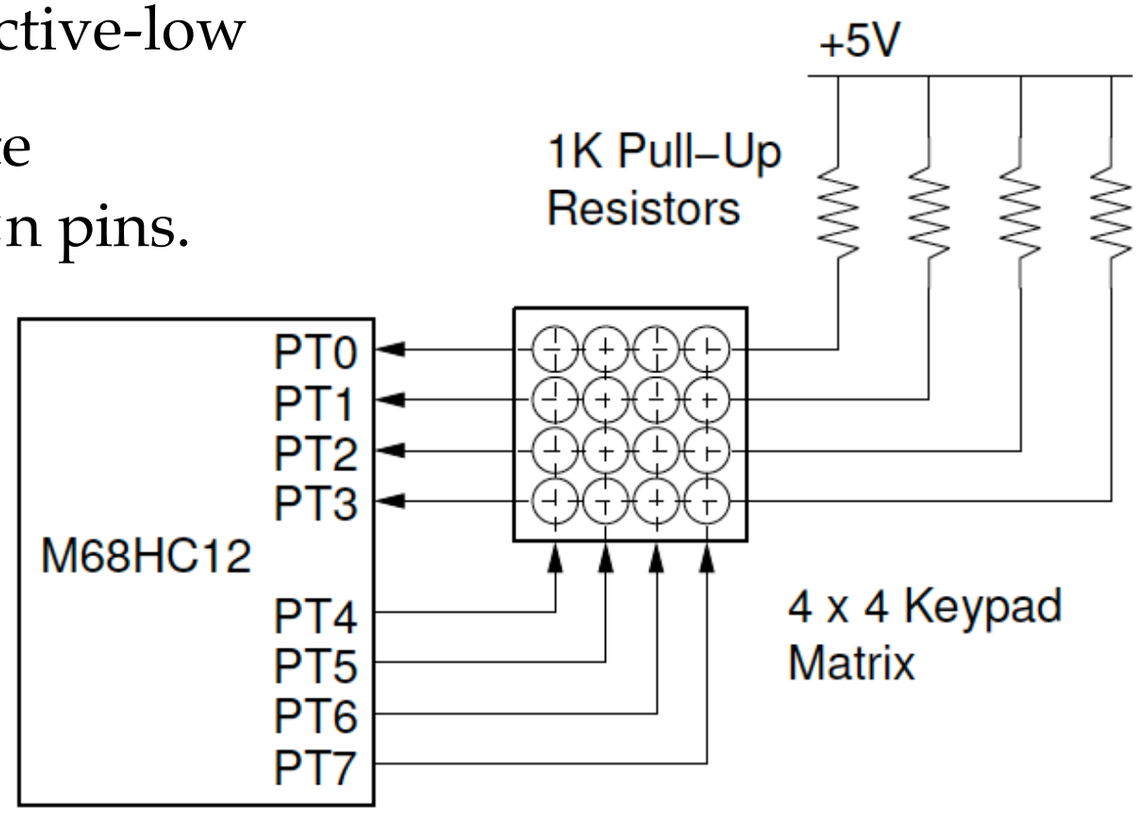
Keypads



Ciaran Moore
ciaran.moore@canterbury.ac.nz

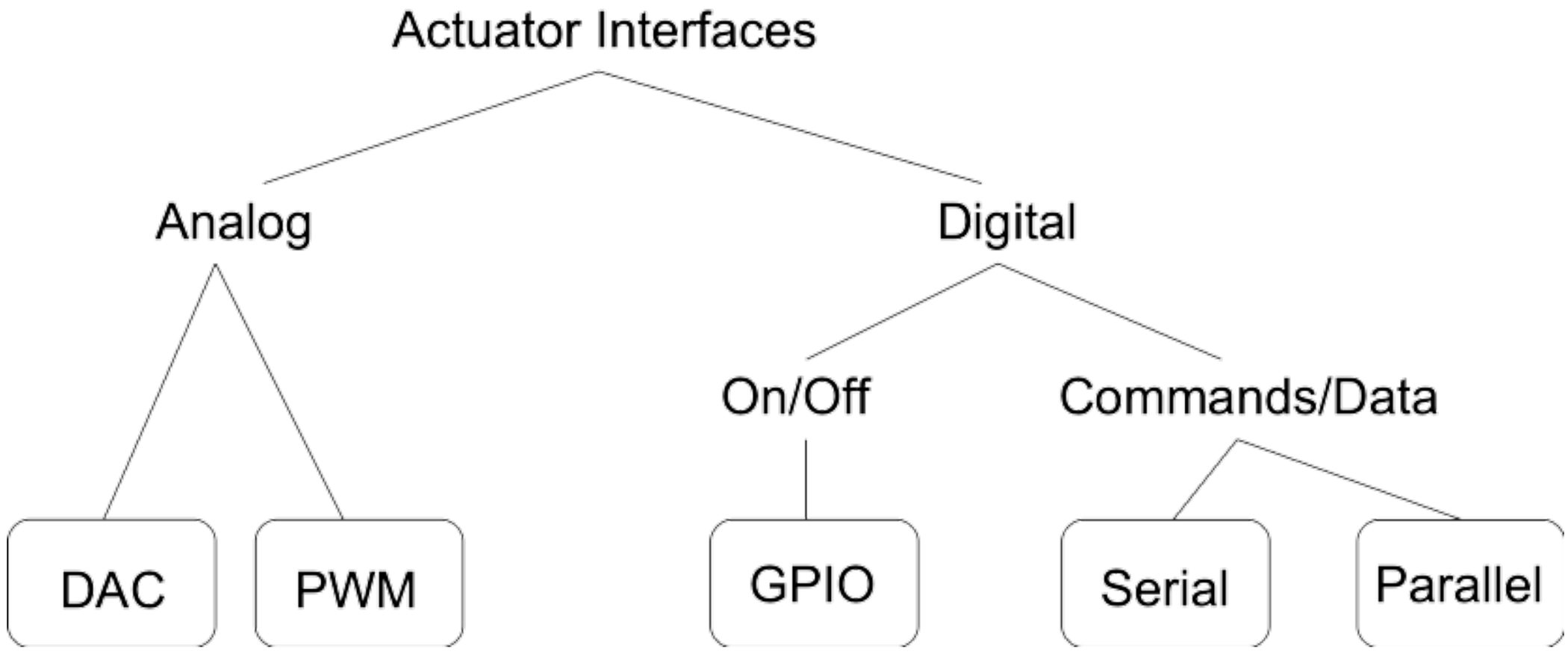
Keypads

- Pin multiplexing
  - Columns are pulsed periodically by outputs PT4-7
  - Rows are read by inputs PT0-3
  - In this example keypresses are active-low
- Combining row and column state allows 2^n keys to be read by 2×n pins.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Outputs



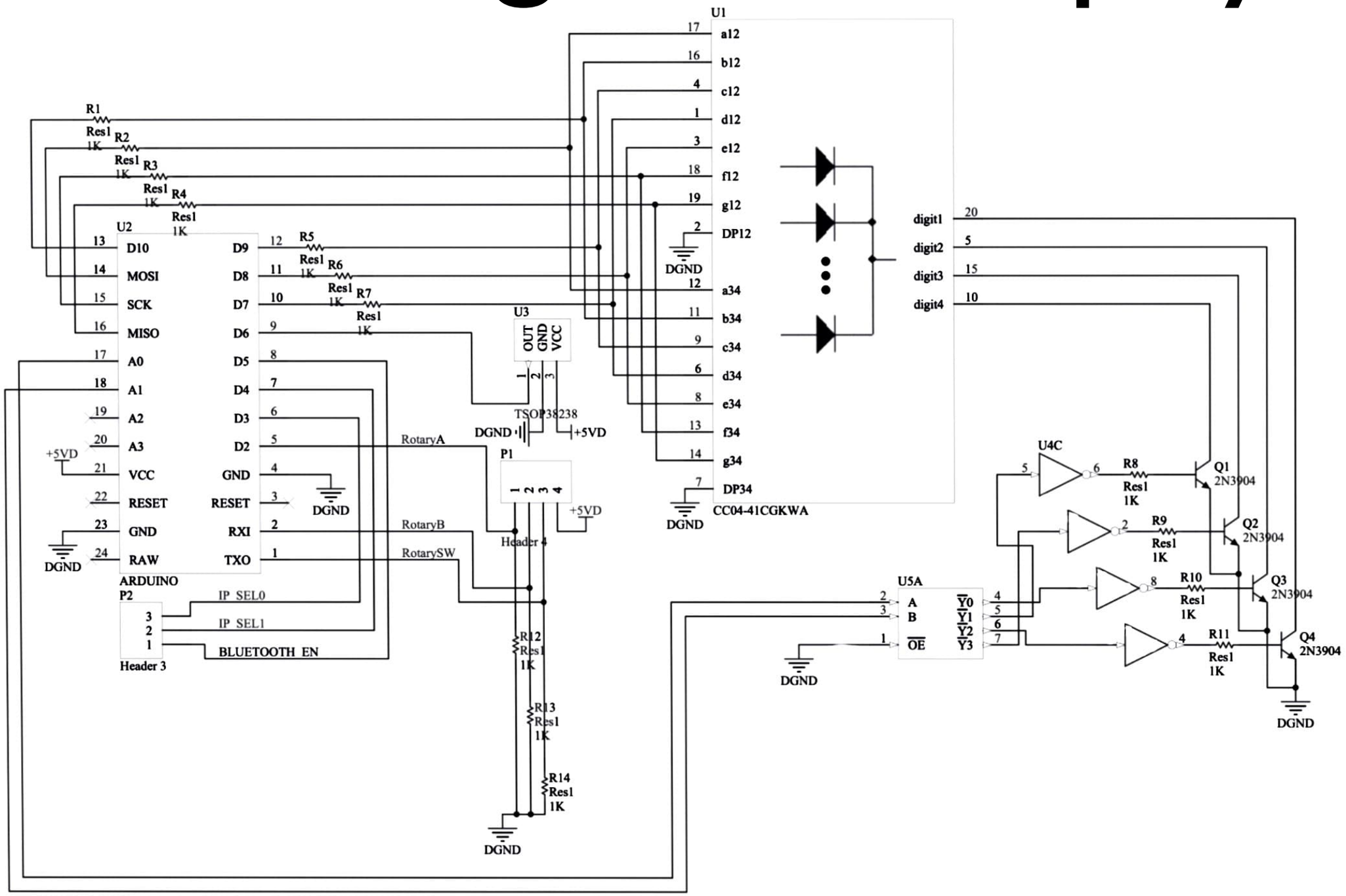
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Outputs

- LEDs
  - Drivers
    - Current-limiting resistors
- Seven-segment displays
- Motors
  - H bridges

Ciaran Moore
ciaran.moore@canterbury.ac.nz

LEDs &
Seven Segment Displays



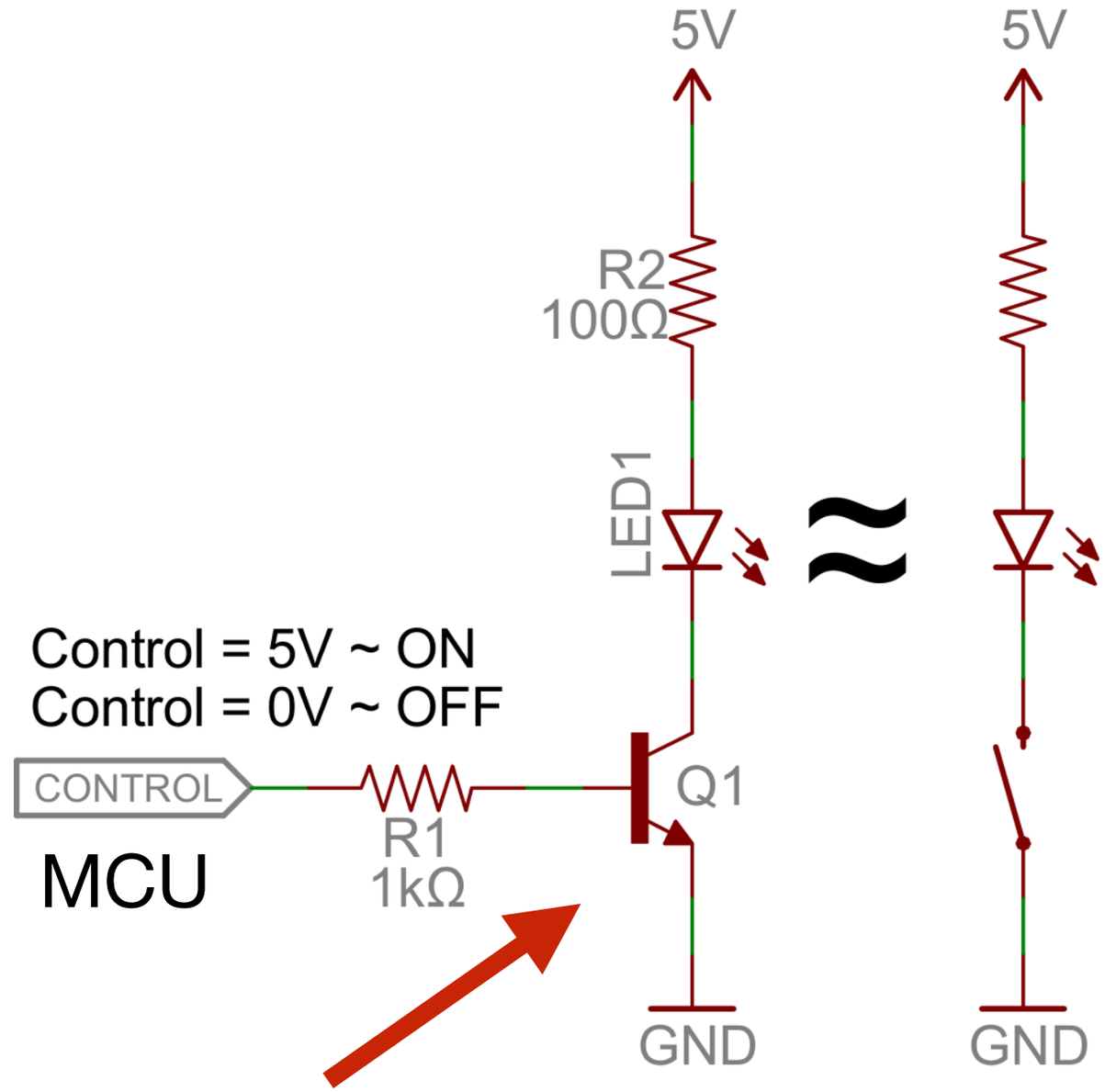
LEDs

- When connecting an MCU to an LED we have two concerns:
  - Can we get enough *current* from the MCU pin? Too little current and the LED will appear dim.
  - Can we limit the *voltage* drop across LED? Too much voltage is like too much lightning: Not Good™.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

LEDs

- LED Drivers
  - MCU pins have limited current supply capability (as low as a few mA).
  - Use MCU pin to control gate (base) of a transistor, which can cope with much higher currents.

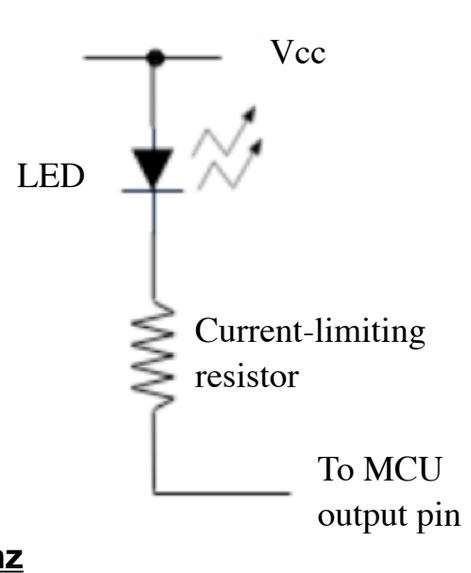


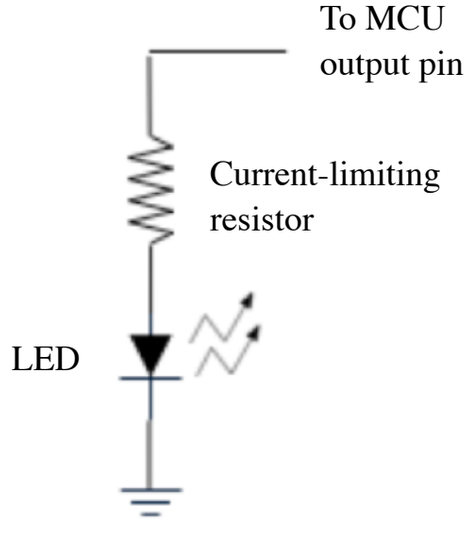
Ciaran Moore
ciaran.moore@canterbury.ac.nz

LEDs

- Current-limiting resistors
  - LEDs have a fixed voltage drop determined by their wavelength (colour).
- Most MCU GPIO runs at 3.3 V or even 5 V – too much for an LED.
- We include series resistors to reduce the voltage across the LED and limit the current through it.

Colour	Wavelength (nm)	Fwd V Drop (V)
Red	610-750	1.8
Yellow	570-590	2.1
Green	500-570	2.2
Blue	450-500	3.2
“White”		3.2



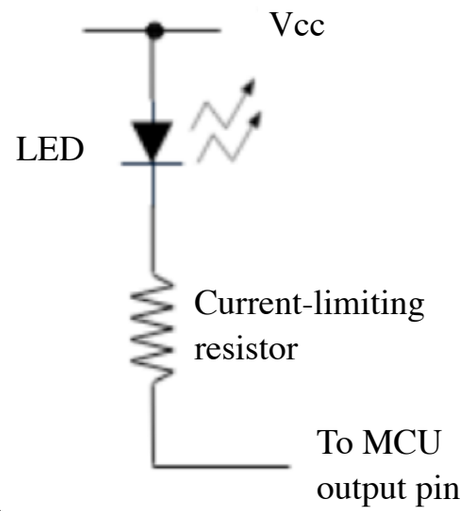


Ciaran Moore
ciaran.moore@canterbury.ac.nz

LEDs

- Current-limiting resistors
  - How big should a current limiting resistor be?

$$V_{CC} = V_{LED} + I_{LED} \times R$$

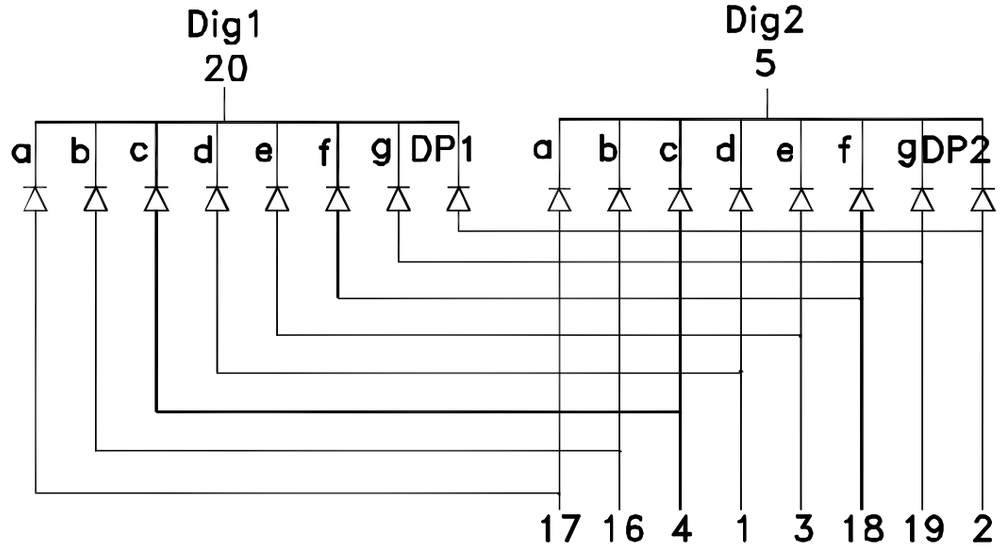


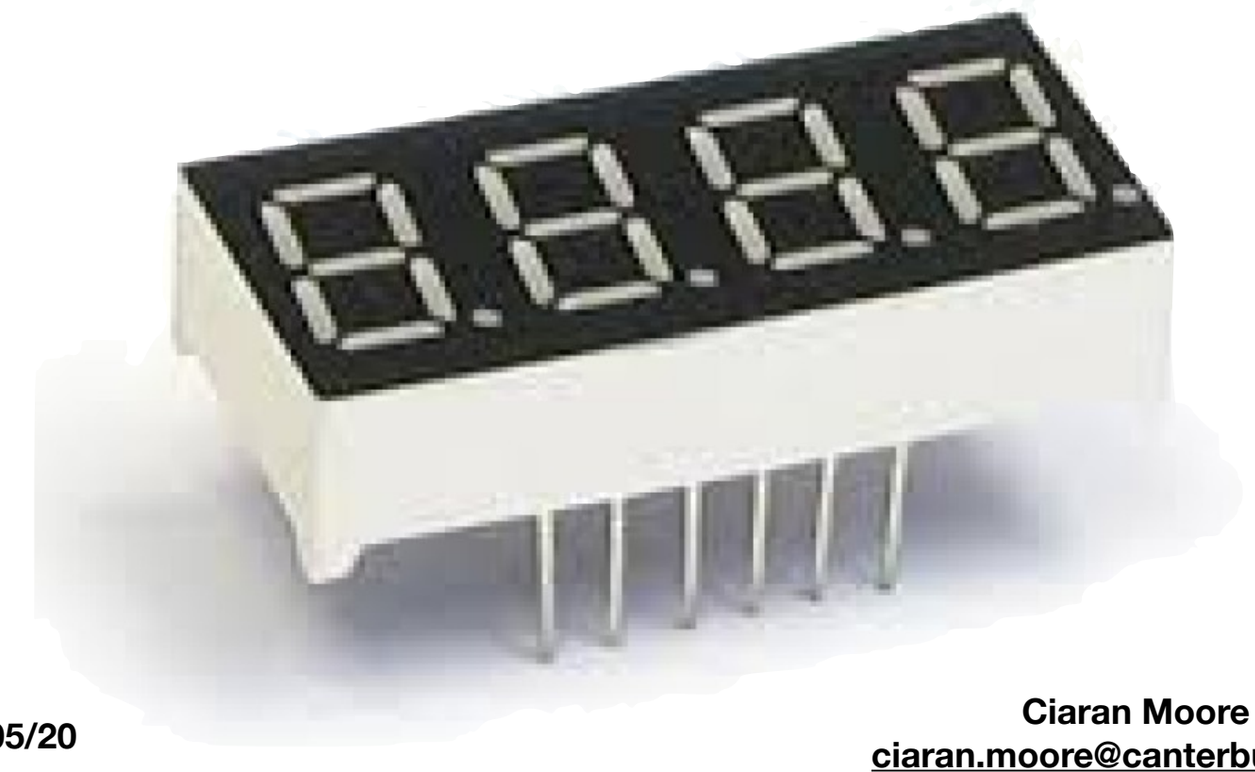
- Choose R to give $I_{LED} \approx$ *10-20* mA*

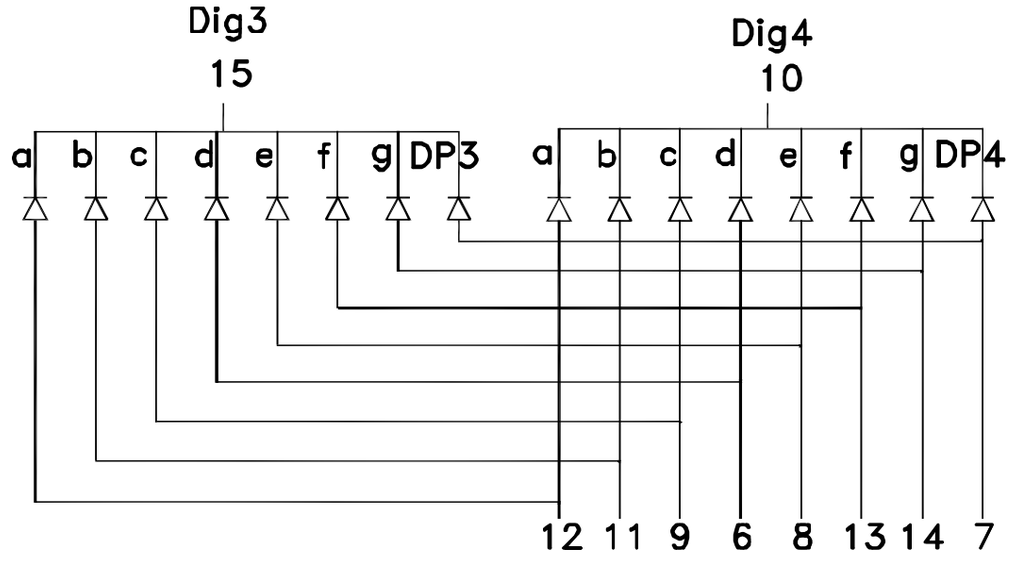
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Seven Segment Displays

- Pin multiplexing
- Digits pulsed sequentially







Ciaran Moore
ciaran.moore@canterbury.ac.nz

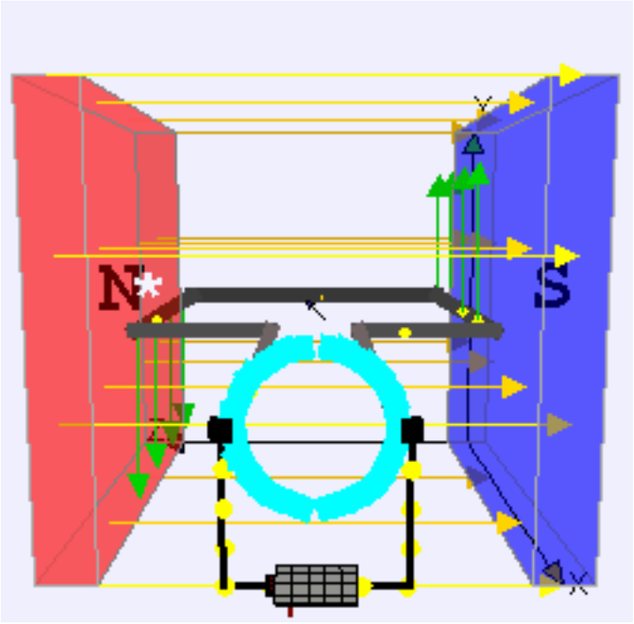
Motors

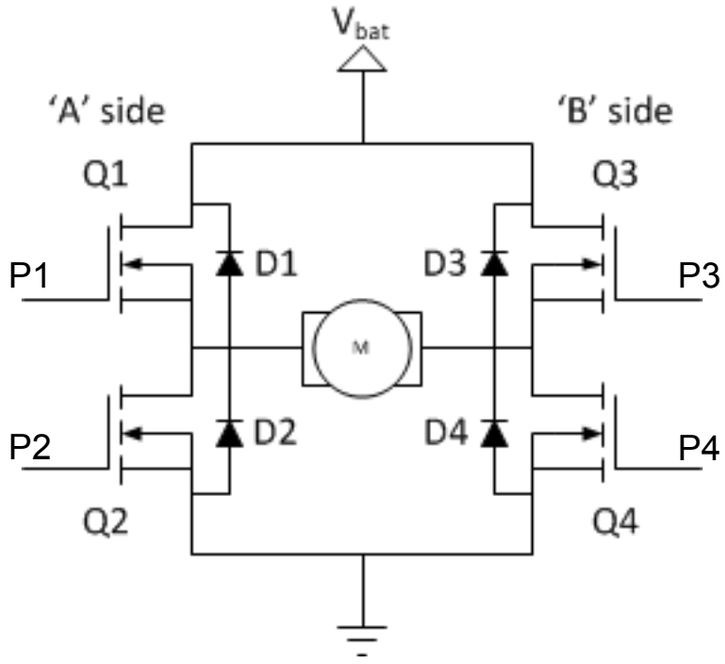
- Motors typically require much more current than an MCU can provide. They also require careful sequencing so that they can rotate clockwise and anticlockwise or in fixed increments.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Motors

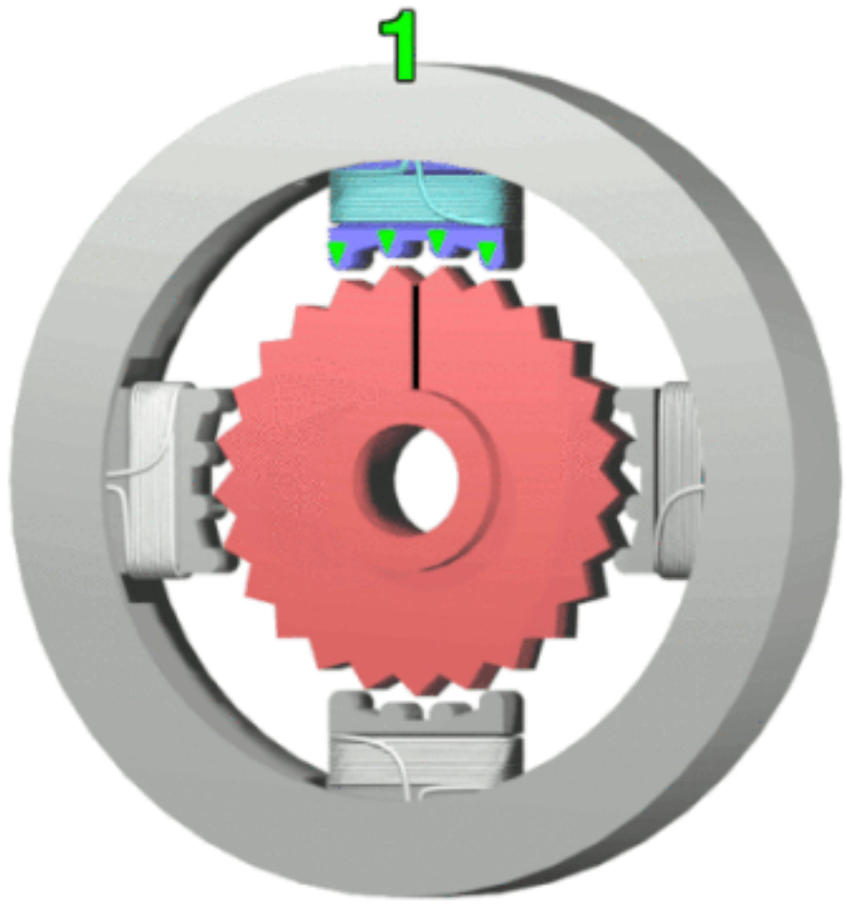
- DC motors
  - The direction of rotation for a DC motor can be swapped by changing the polarity of its connections.
  - An H-bridge dynamically changes the polarity *and* copes with the high currents the motor needs.
- For forward motion pins P1 and P4 are high, so transistors Q1 and Q4 are on.
- For reverse motion pins P3 and P2 are on, which turns on Q3 and Q4.
- Diodes are included to allow the motor coil to discharge when motion is stopped or reversed.



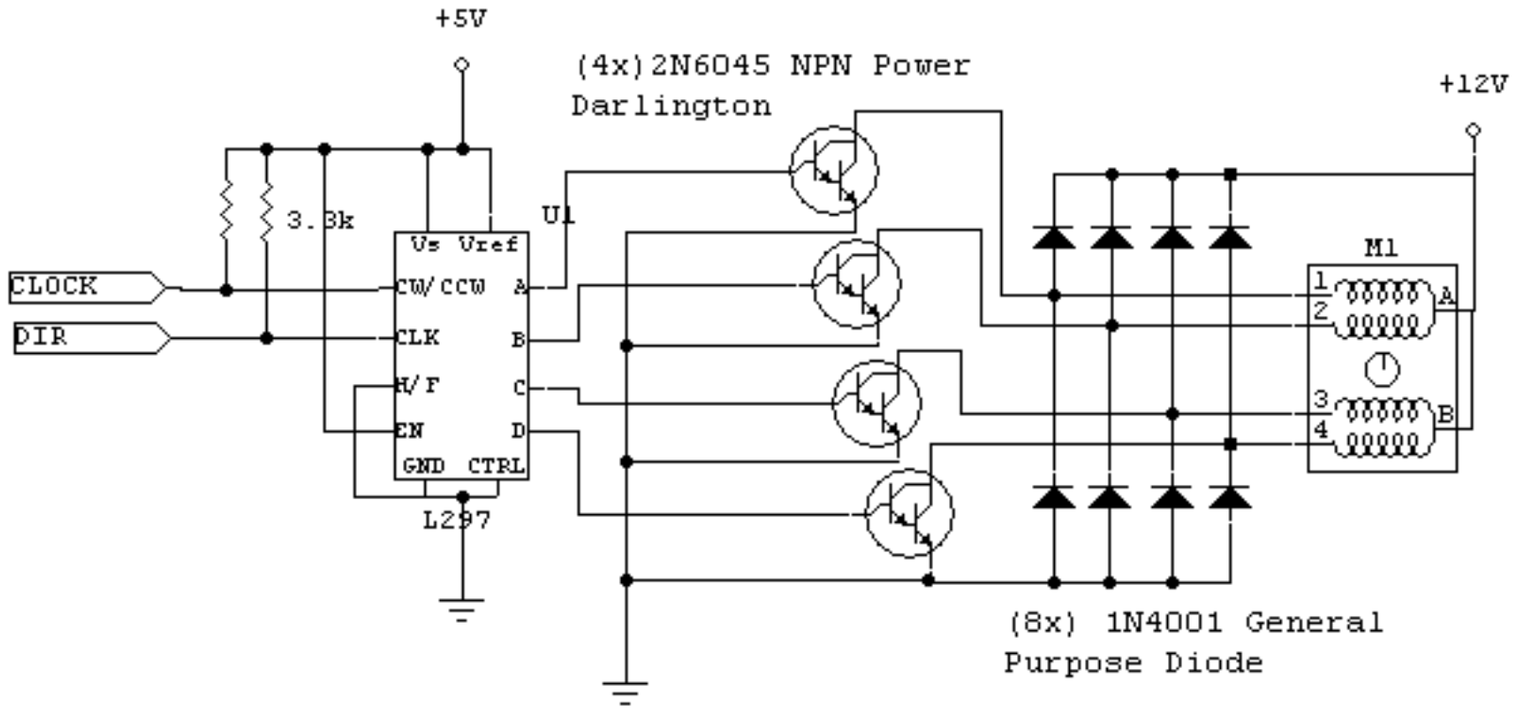


Ciaran Moore
ciaran.moore@canterbury.ac.nz

Motors



- Stepper Motors
  - Stepper motors use multiple electromagnets and a toothed central gear to rotate in fixed increments.
  - Pulsing the electromagnets in sequence leads to continuous rotation.
  - Often a stepper motor controller (e.g. L297) is used to simplify the interface between the motor and the MCU.
  - Like DC motors, stepper motors cannot be driven directly by MCU GPIO pins. Instead, they use transistors to act as high current switches controlled by low current pins.



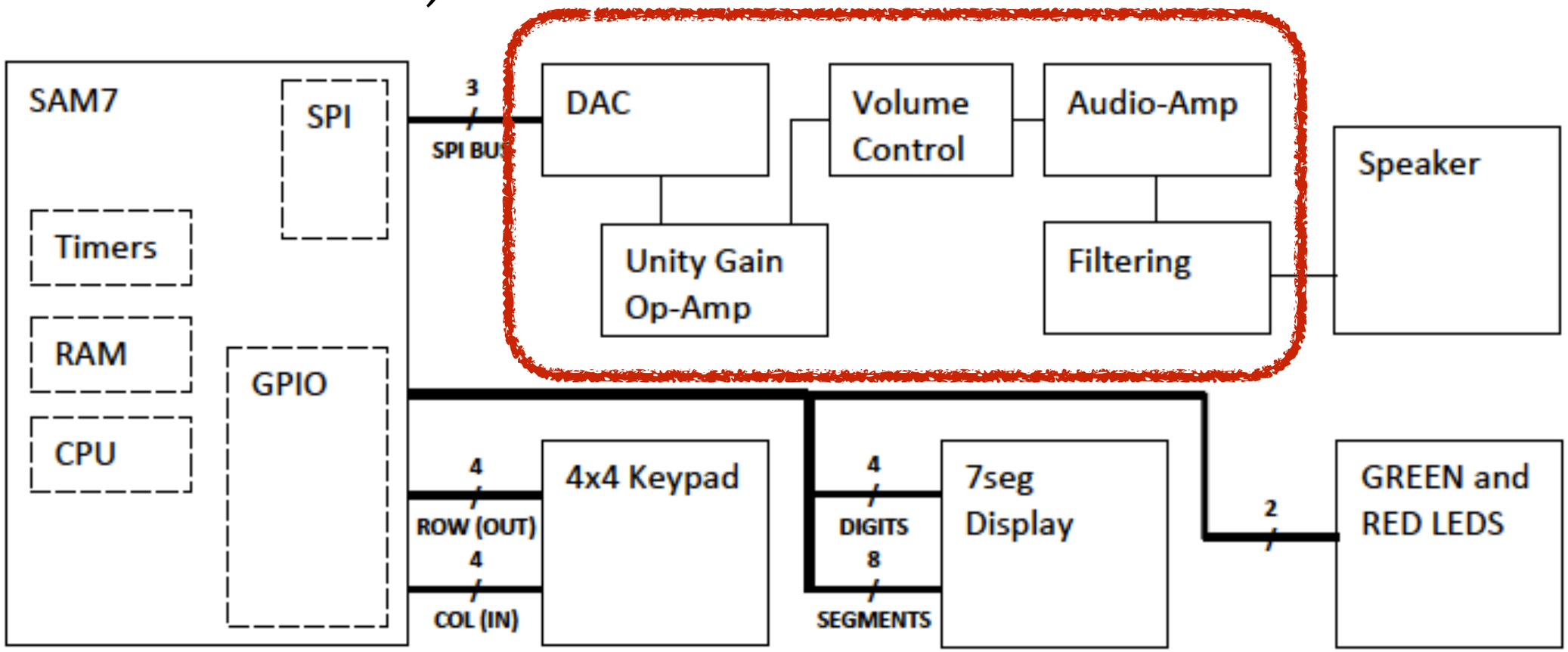
Signal Conditioning

- A common theme with sensors and actuators is that they are seldom able to be connected directly to an MCU.
- Instead, additional components to alter voltage, current, biasing, impedance, frequency, etc. are often required.
- These signal conditioning components can become a large part of your design.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Signal Conditioning

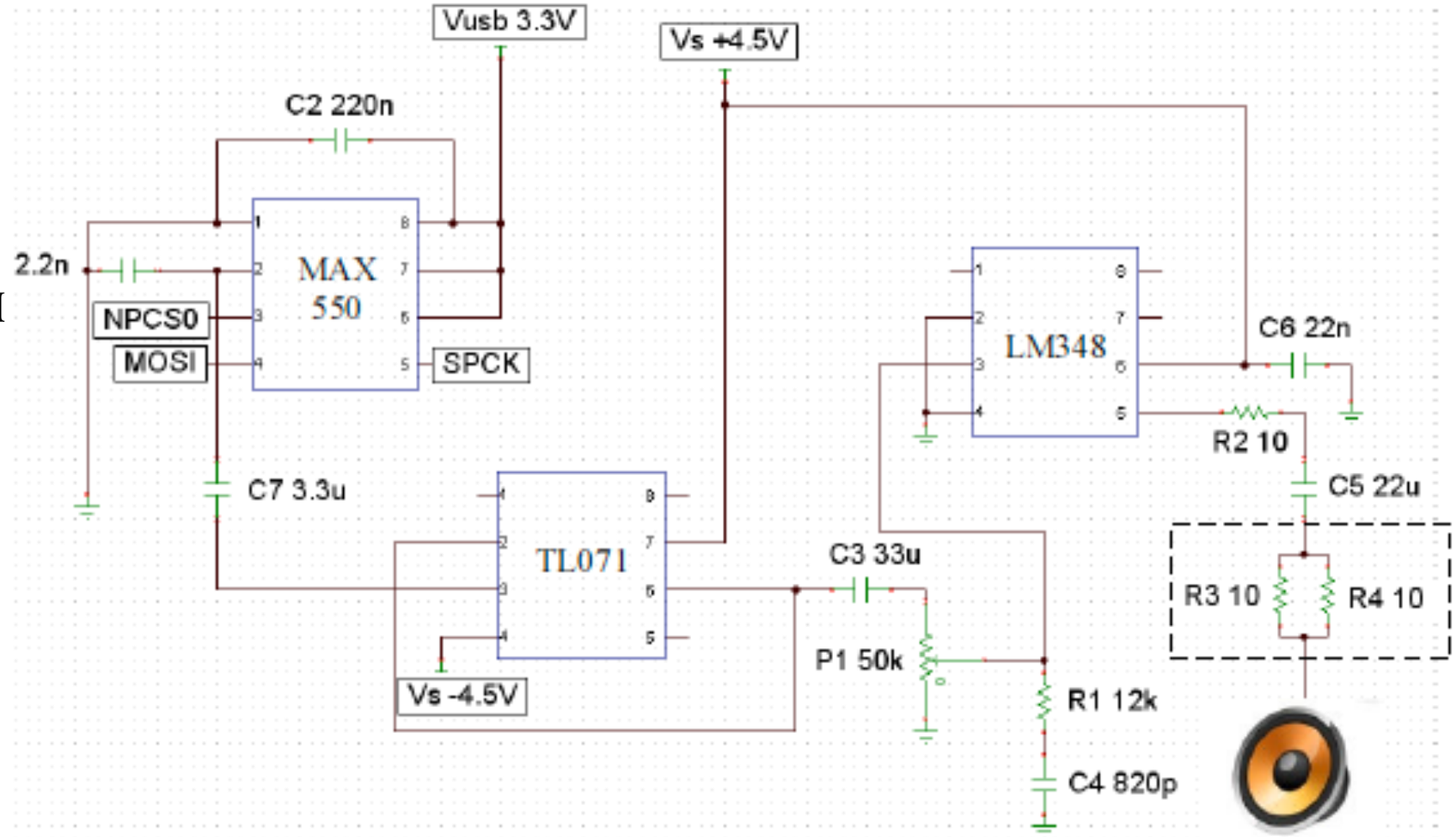
- Example: Electronic jukebox (ENEL353, ~2004-08)



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Signal Conditioning

From MCU
SPI port



DAC Pre-Amplifier Power amplifier Actuator (speaker)

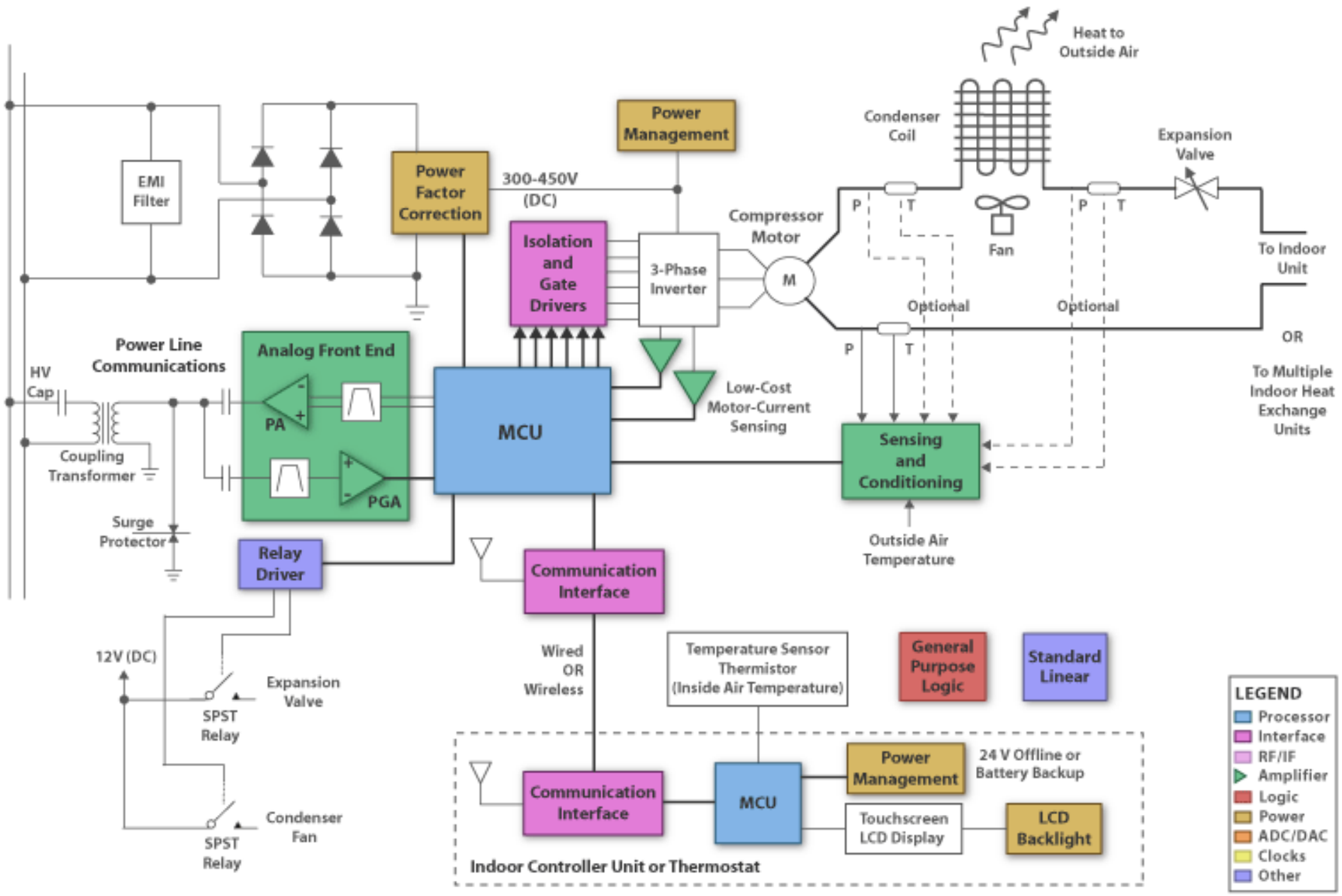
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Reference Designs

- How do we know what sort of signal conditioning hardware to include? How do we know how large the current-limiting resistors should be? What about the time constant for an RC debouncing circuit?
- Fortunately, manufacturers typically supply *reference designs* to make it easier to use their components.
- *Application notes* explain the reasoning used in the reference designs.
- *Data sheets* contain key facts and measurements about components.
- Although a reference design may not be completely suitable for your finished product, they provide a starting point from which to iterate.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Home heating and ventilation system reference design



Ciaran Moore
ciaran.moore@canterbury.ac.nz

*http://www.ti.com/solution/hvac_heating_ventilating_and_air_conditioning

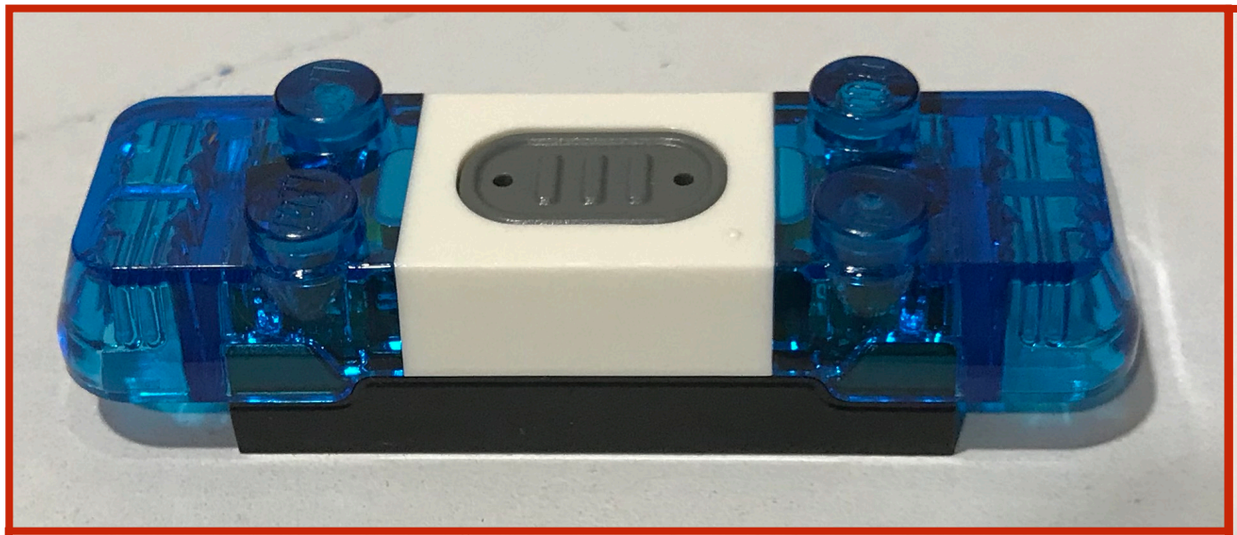
Memory I

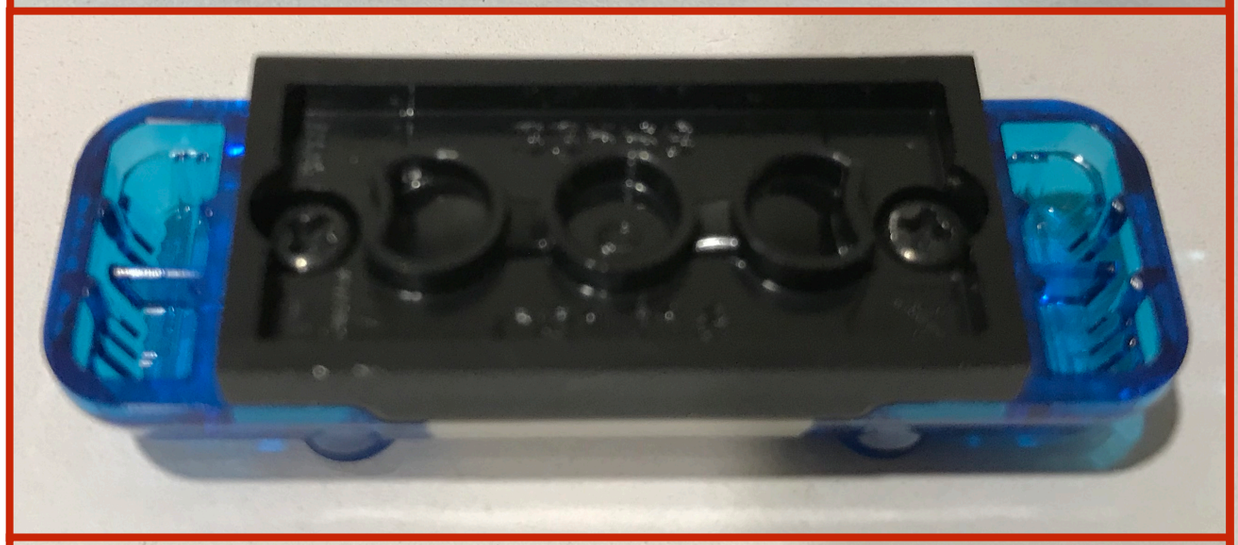
ENCE361: Design & Architecture: Lecture Block 2

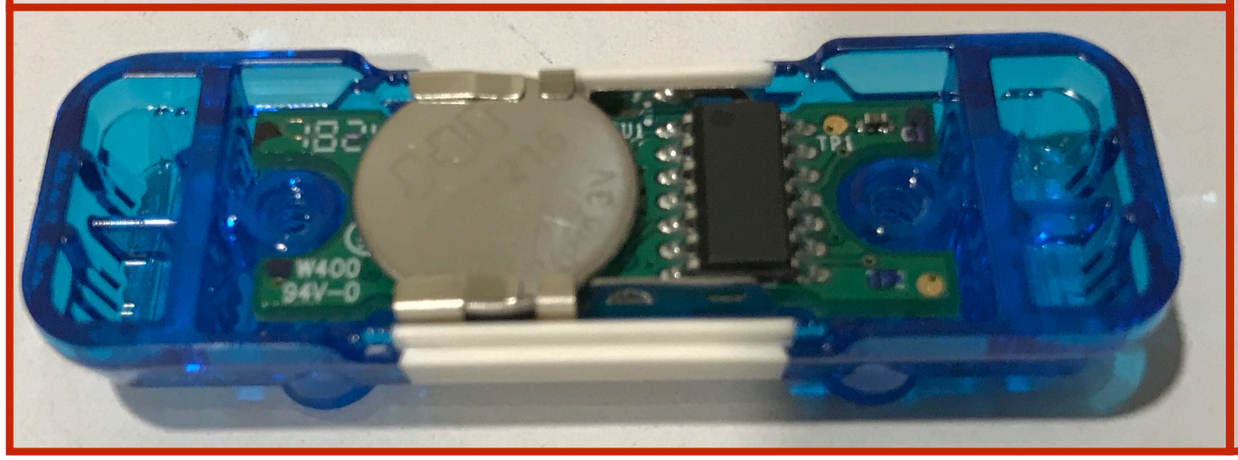

LEGO
CITY
Ages/edades
6+
60209
Sky Police
Diamond Heist
400
pcs/pzs
Building Toy
Jouet de construction
Juguete para Construir
POLICE
battery
included
LIGHT & SOUND

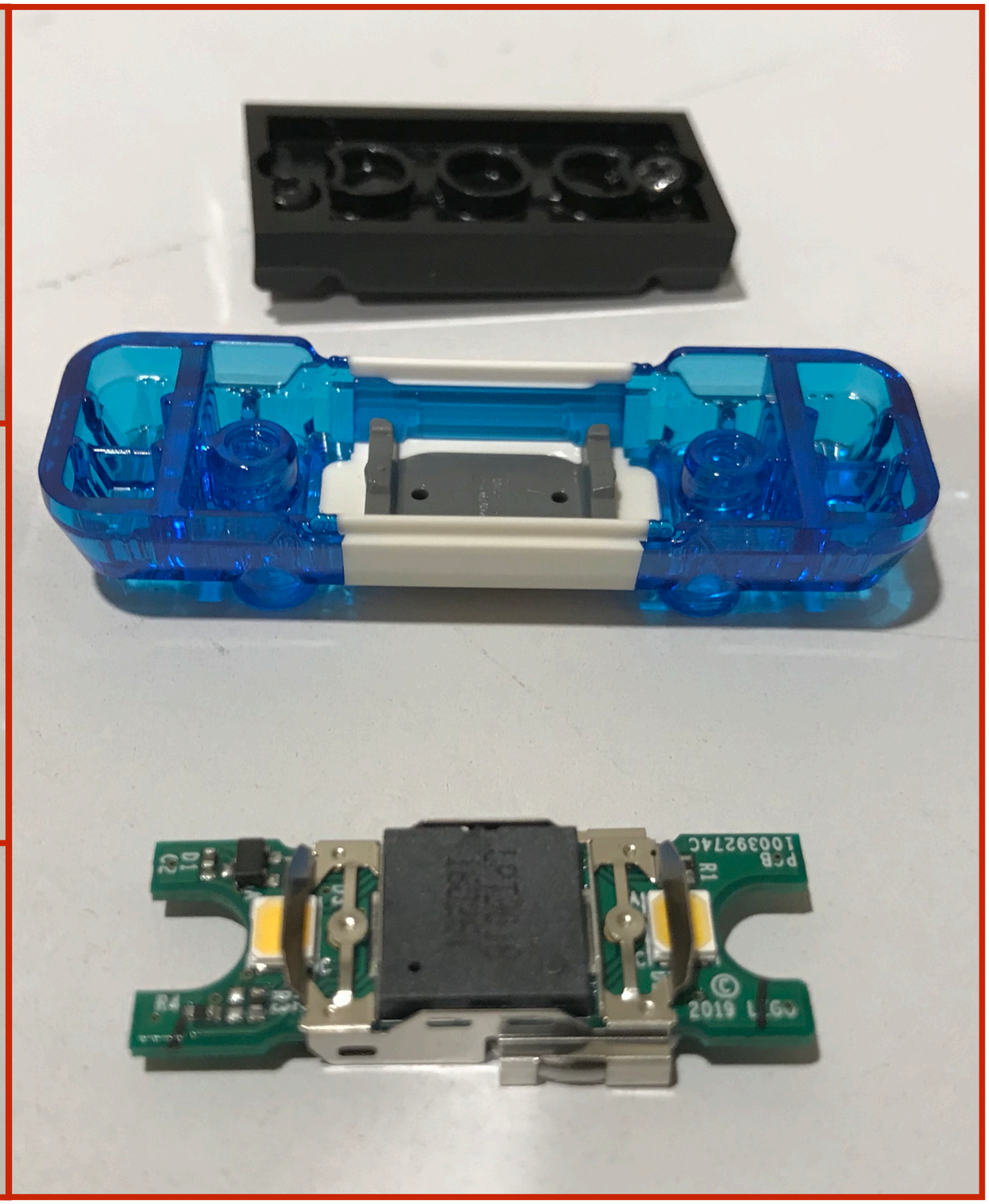

POLICE


POLICE
POLICE









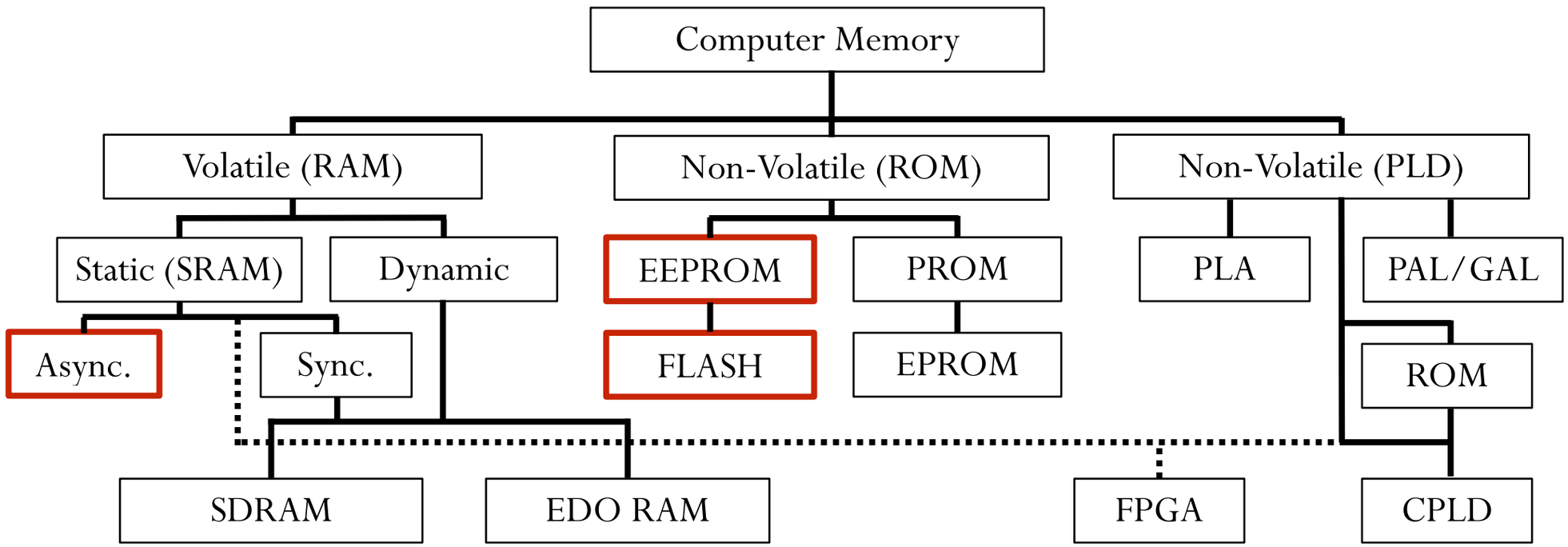
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Outline

- (S)RAM
- EEPROM
- FLASH

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Taxonomy



Note 1: This is not an absolute list

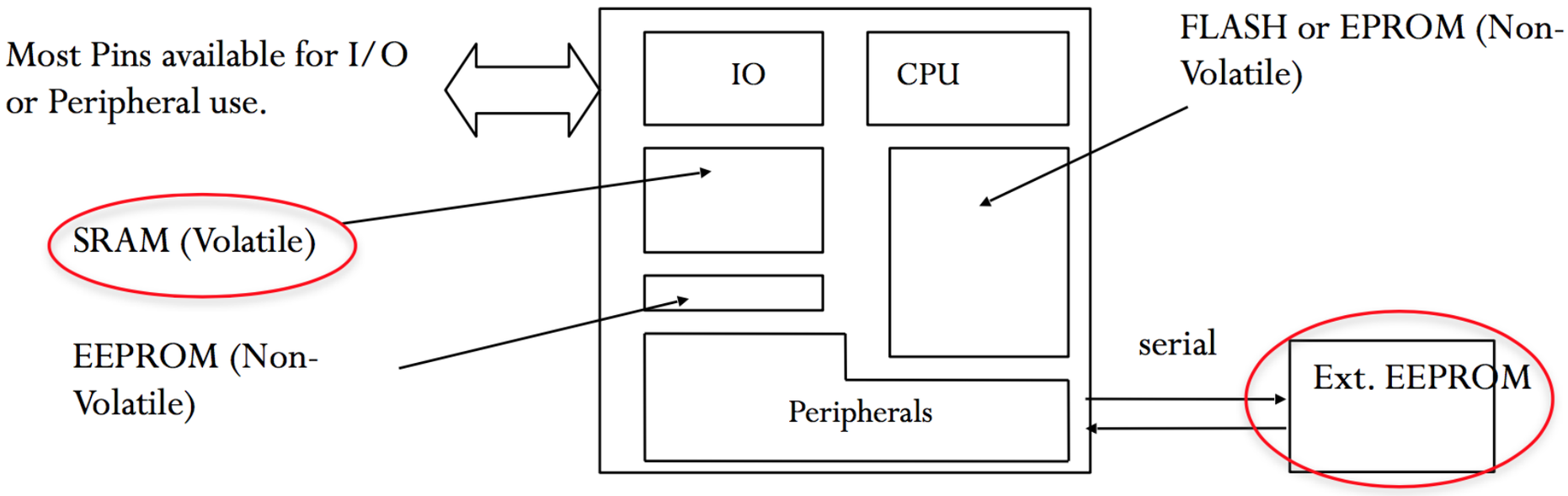
Note 2: Primary acronyms are defined in the glossary

Prominent in modern embedded systems

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Microprocessor Memory Interfacing

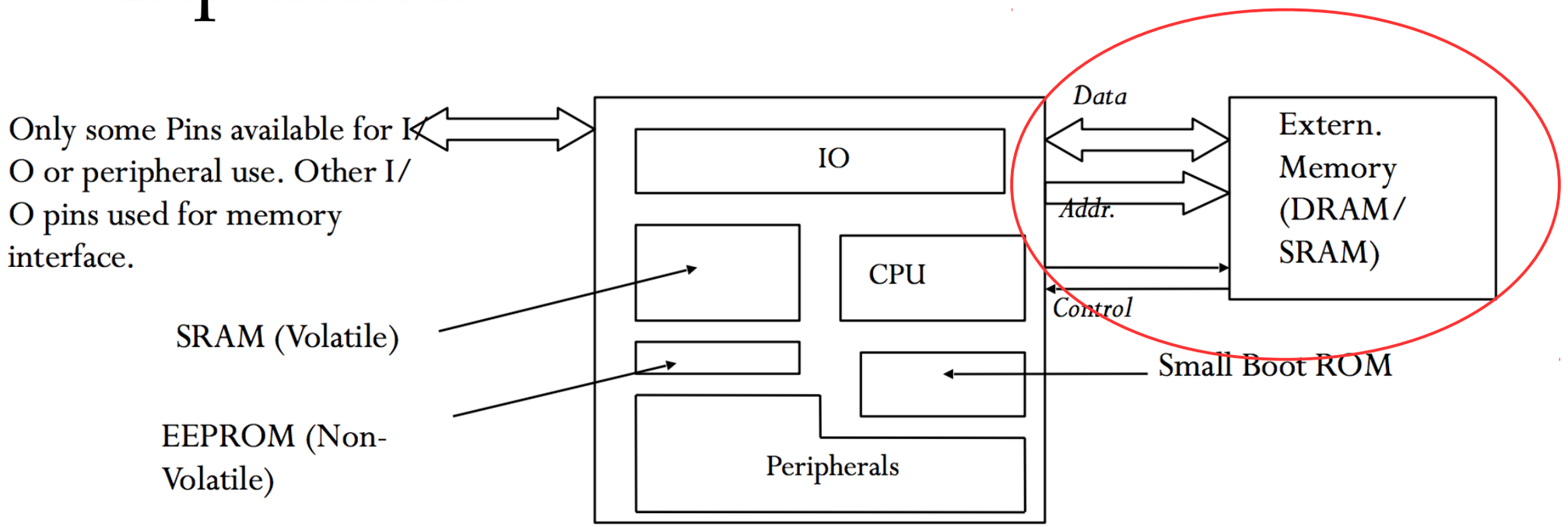
- Single Chip



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Microprocessor Memory Interfacing

- Single Chip
- Expanded



Note: Address and Data Buses can be shared (multiplexed) or not shared (non-multiplexed).

Ciaran Moore
ciaran.moore@canterbury.ac.nz

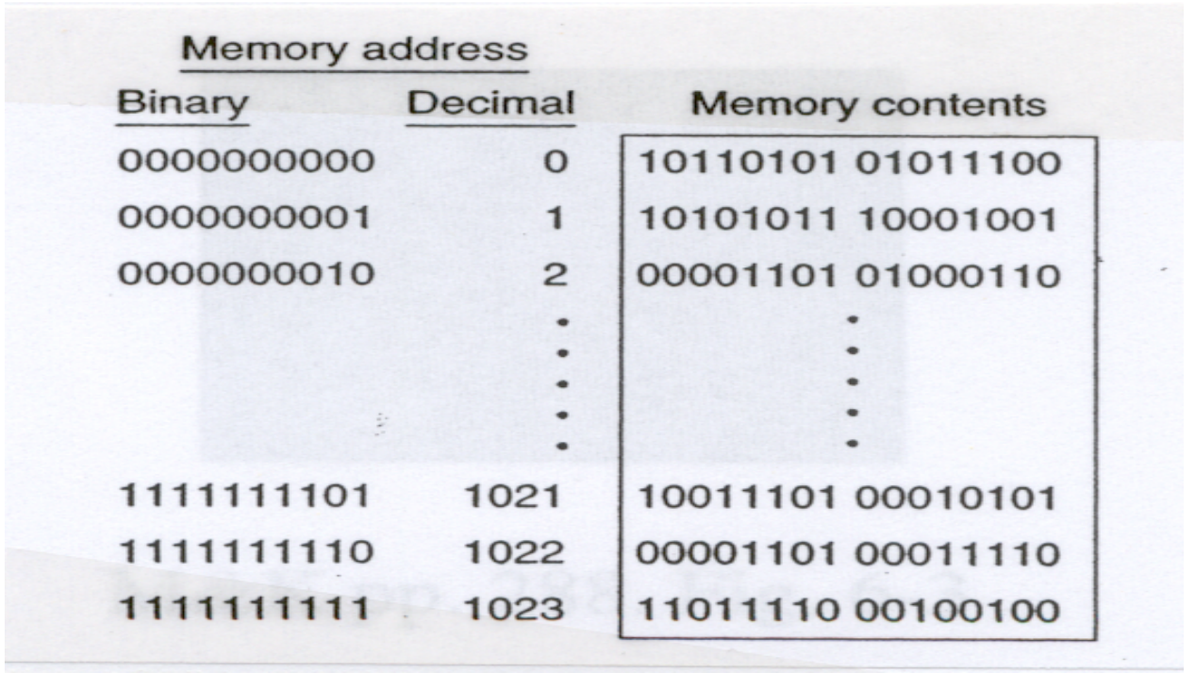
Microprocessor Memory Interfacing

- Single Chip
- Expanded
  - Larger memory space
  - Fewer free I/O pins

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Random Access Memory (RAM)

- RAM is traditionally asynchronous
- No location dependence on read/write time: data takes ~the same amount of time to access.

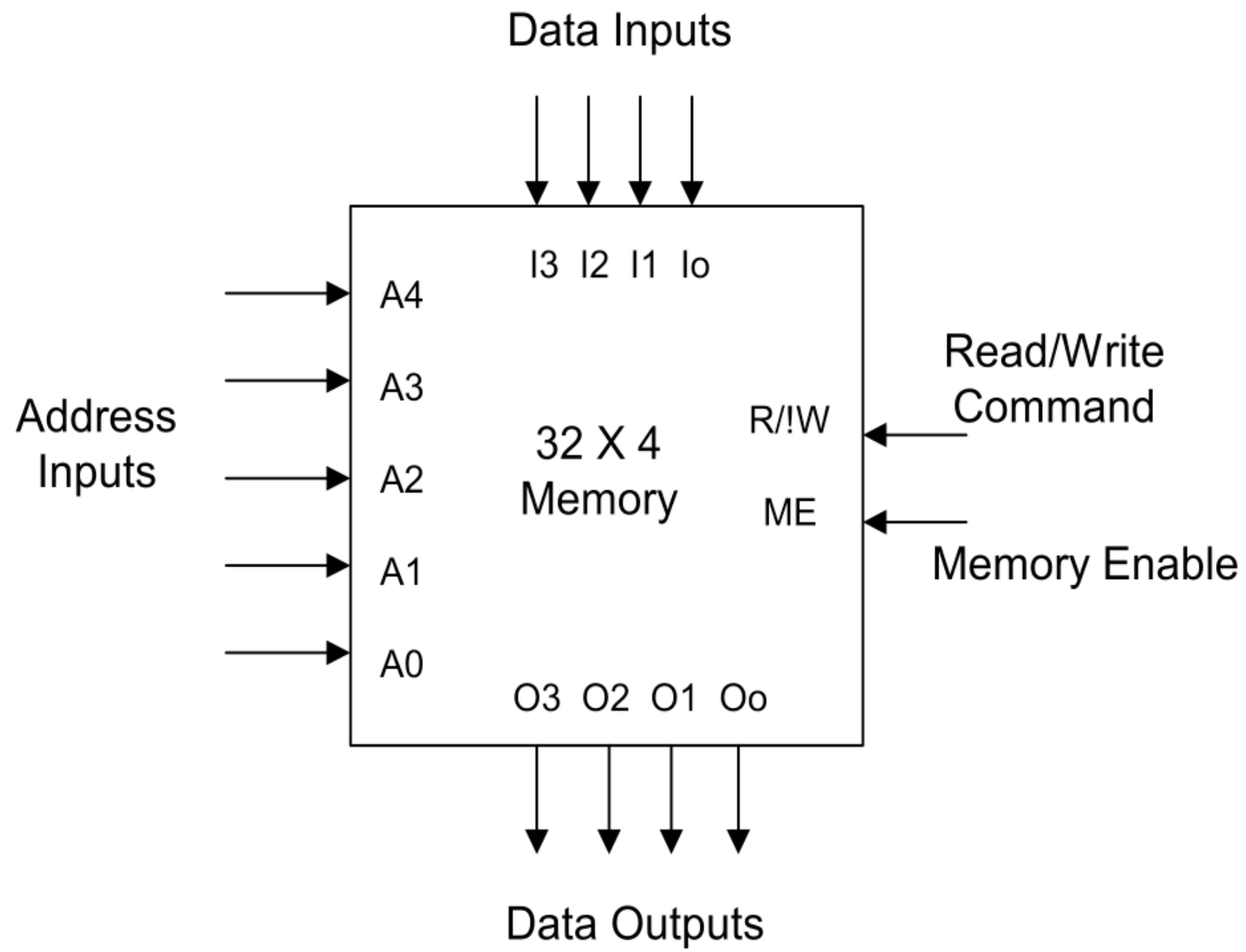


Memory address		
Binary	Decimal	Memory contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
⋮	⋮	⋮
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

- In this example, only 10 bits are required to address a 1024 *word* memory array.
- Internally memory is configured as 16 bit data. So the capacity of this device is 16×1024 = 16,384 *bits*.
- The data word size of memory is independent of the address range.

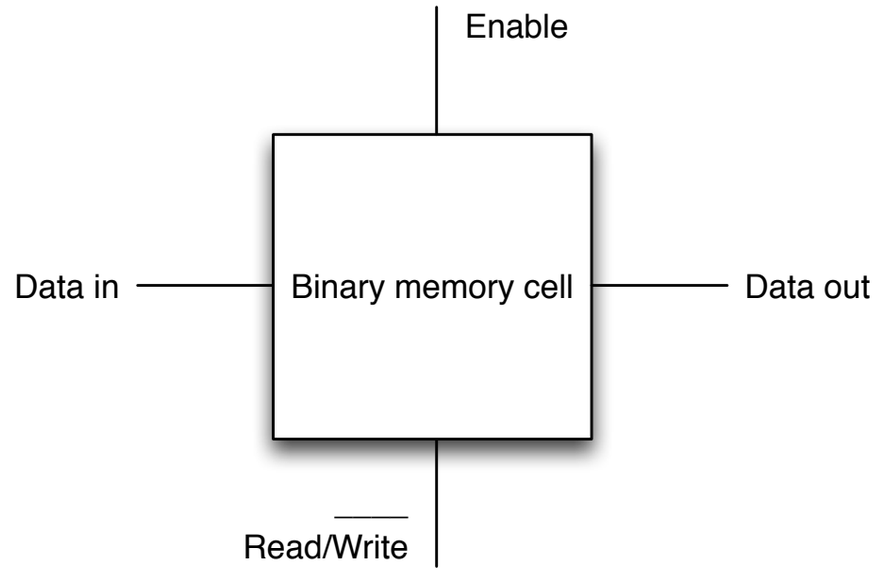
Ciaran Moore
ciaran.moore@canterbury.ac.nz

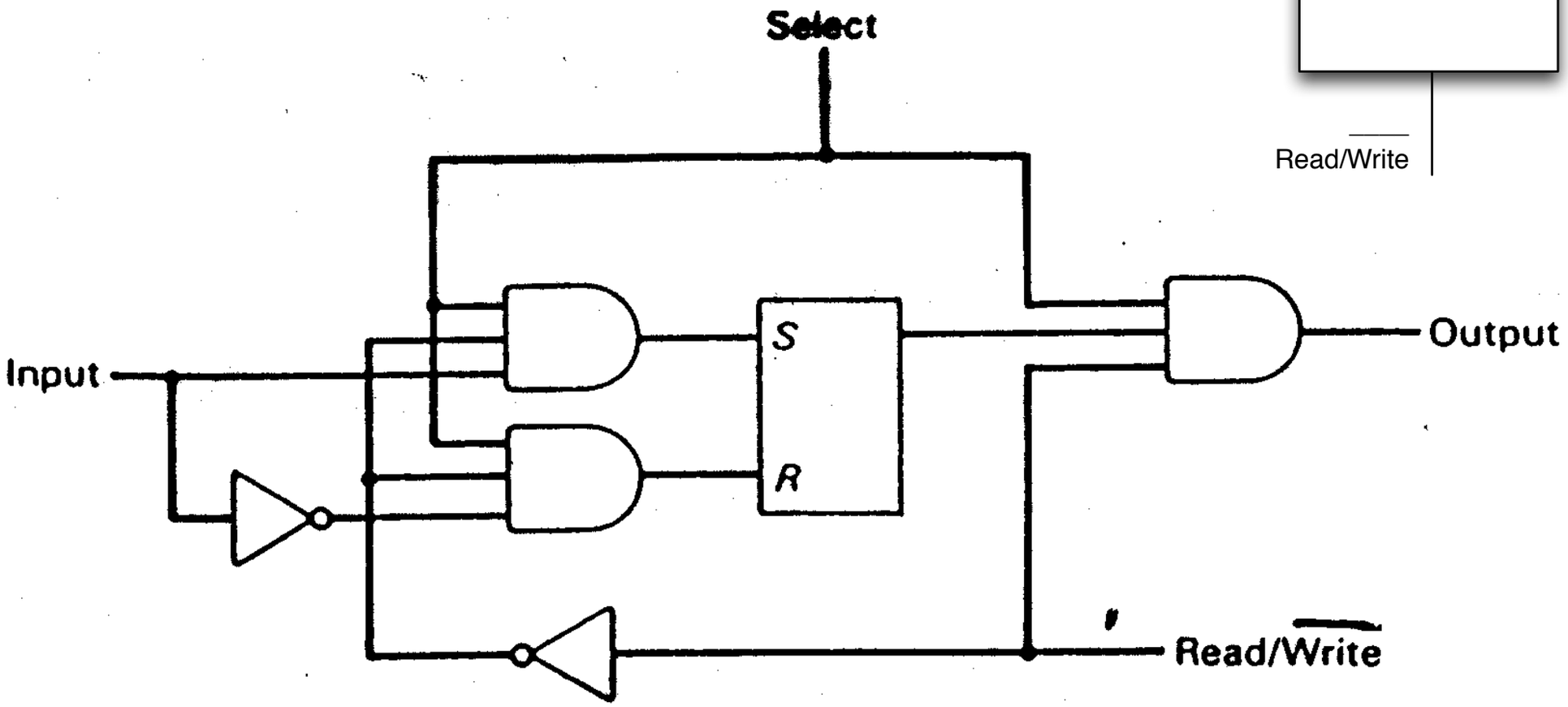
Static RAM



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Binary Memory Cell





Mano, 1984

Ciaran Moore
ciaran.moore@canterbury.ac.nz

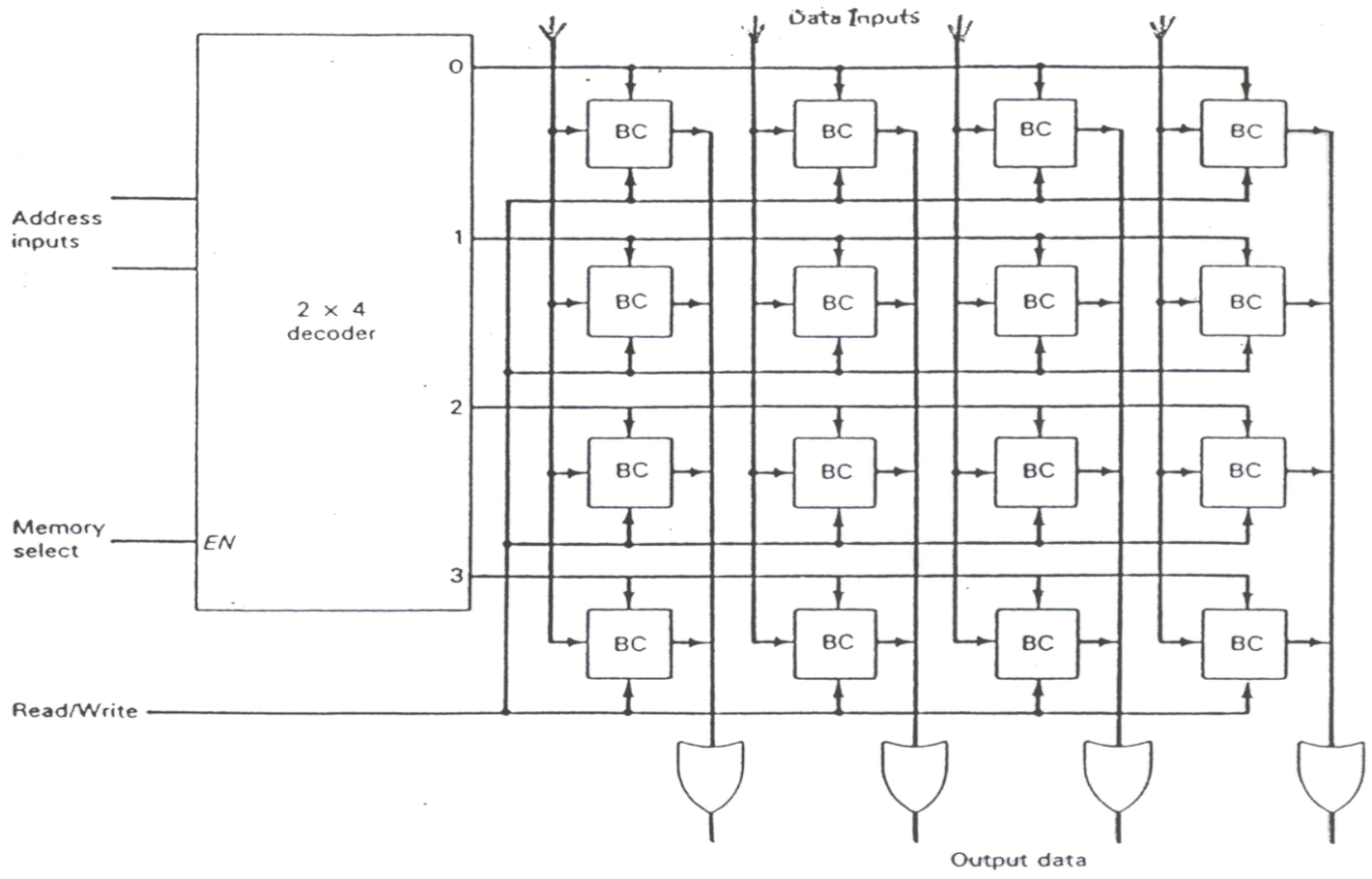
Binary Memory Cell

- The *read*/*write* control input is used for data flow.
- The *enable* control asserts the binary cell.
- The SR latch is either *set* or *reset* by *data in* when *read*/*write* and *enable* are both TRUE.
- *Data out* is read out of the binary cell when *read*/*write* is FALSE and *enable* is TRUE.
- Operations are asynchronous, i.e. no clock is used.
- Once functionality is defined, the binary cell can be thought of as a module.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

SRAM Modular Construction

- Using interconnected binary cells



Mano, 1984

Ciaran Moore
ciaran.moore@canterbury.ac.nz

2D Decoders

- Improve memory access time

FIGURE 6-7
Diagram of a 4 × 4 RAM

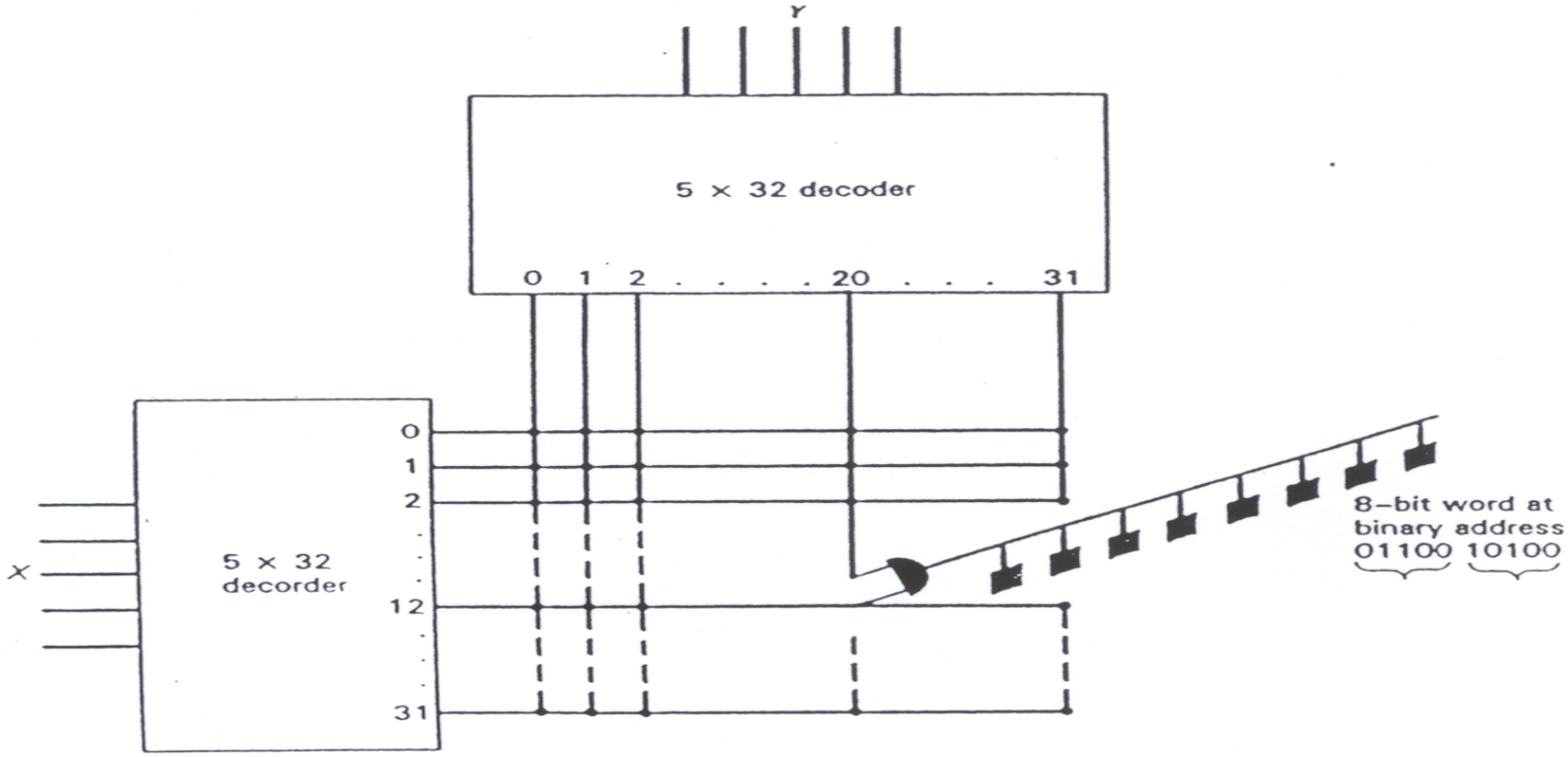


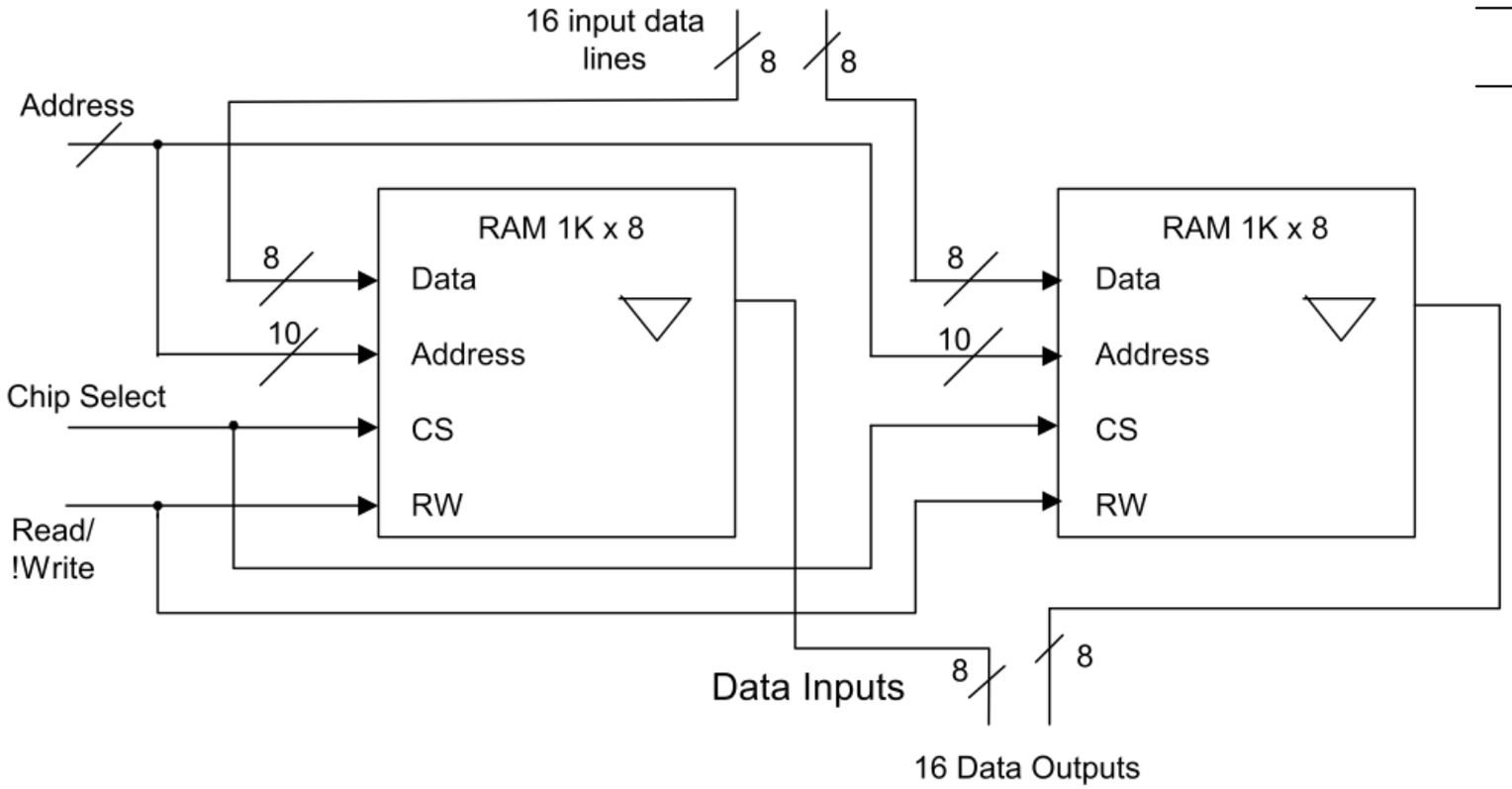
FIGURE 6-8
Two Dimensional Decoding Structure for a 1K × 8 RAM

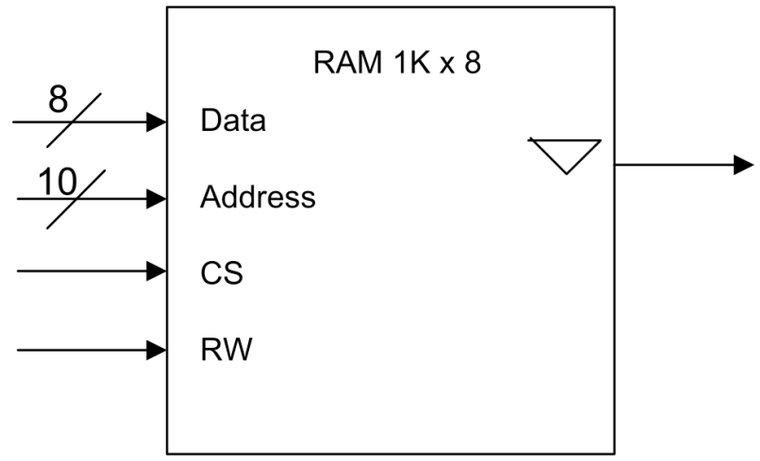
Mano, 1984

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Interfacing & Expansion

- Double word size:





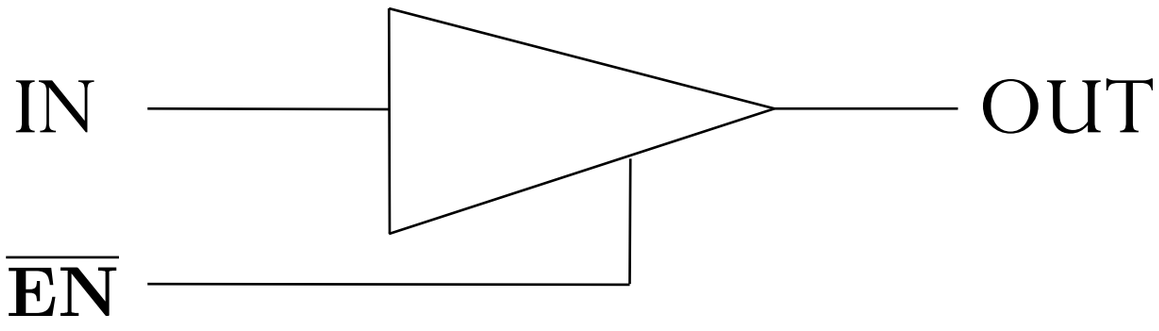
Block diagram of a 1K x 8 RAM Chip

- Pin count gets high quickly.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Revision on Tri-State Buffers

- Tri-state buffers have three states: logic 1, logic 0, and **high impedance** (hi-Z); i.e. V_H, V_L, or an *open circuit*.



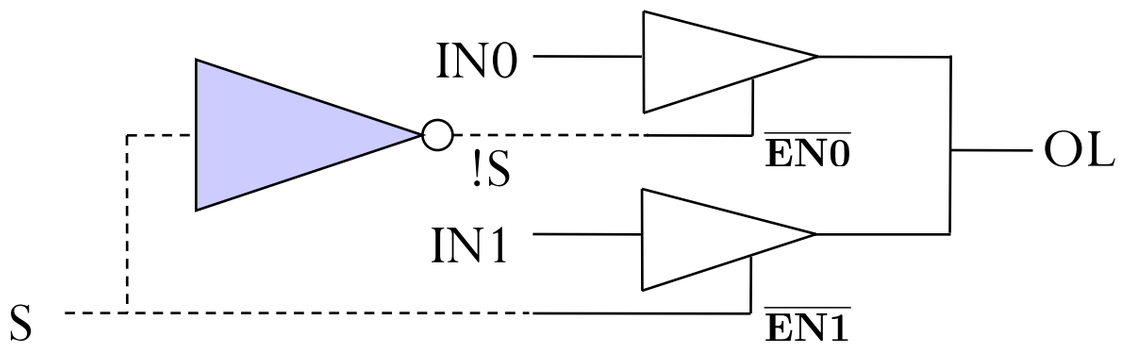
EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

Mano & Kime, 2000

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Revision on Tri-State Buffers

- Tri-state buffers are used to drive common output lines



EN1	EN0	IN1	IN0	OL
0	0	X	X	Hi-Z
S = 0	!S =1	X	0	0
S = 0	!S =1	X	1	1
S = 1	!S =0	0	X	0
S = 1	!S =0	1	X	1
1	1	0	0	0
1	1	1	1	1
1	1	0	1	?
1	1	1	0	?

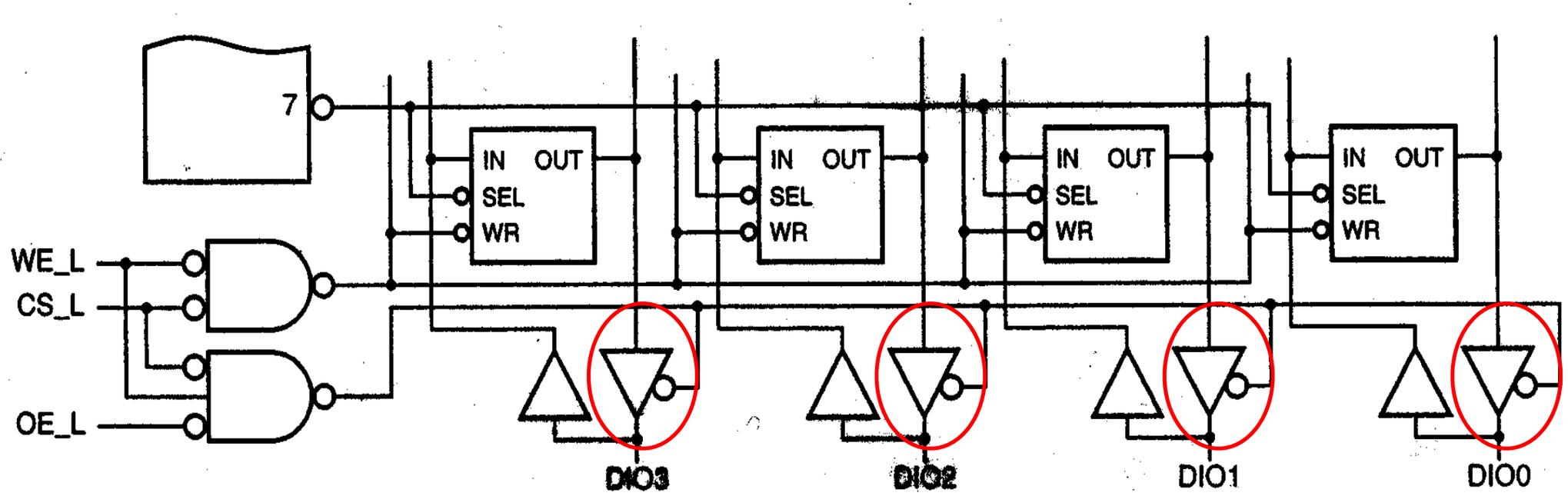
Mano & Kime, 2000

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Interfacing & Expansion

- Save pins by combining input & output pins using tri-state buffers

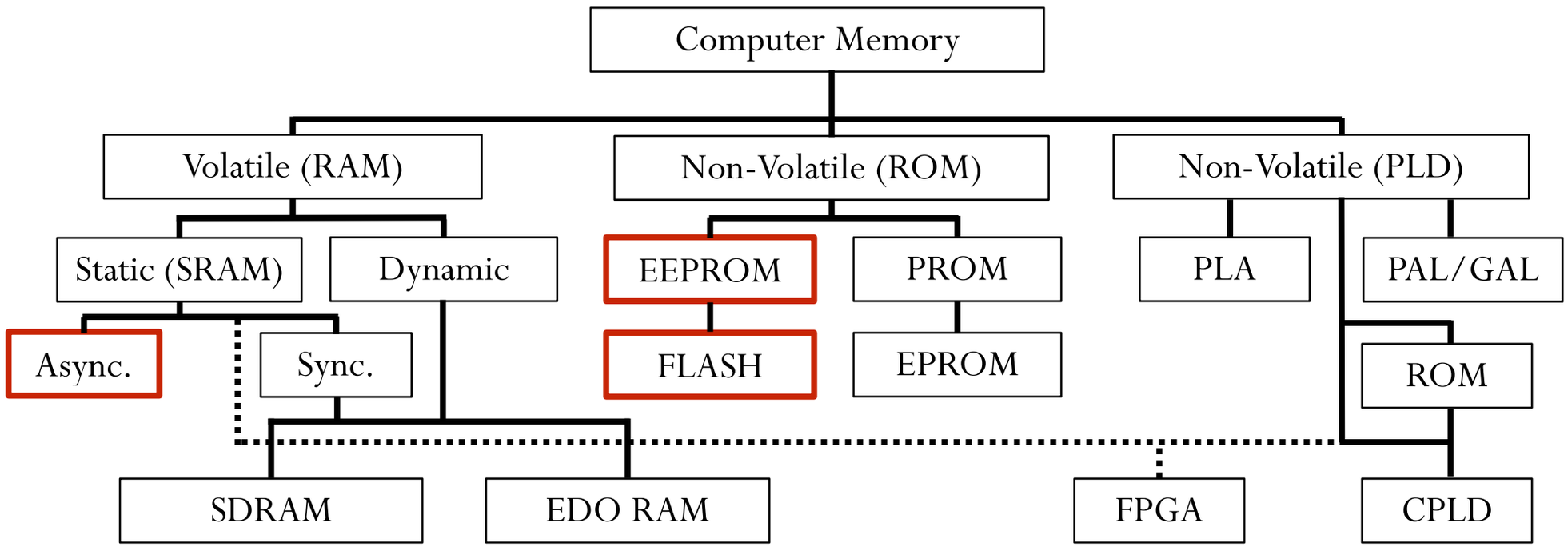
Output-buffer control in an SRAM with a bidirectional data bus.



Wakerly, 2000

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Taxonomy



Note 1: This is not an absolute list

Note 2: Primary acronyms are defined in the glossary

Prominent in modern embedded systems

Ciaran Moore
ciaran.moore@canterbury.ac.nz

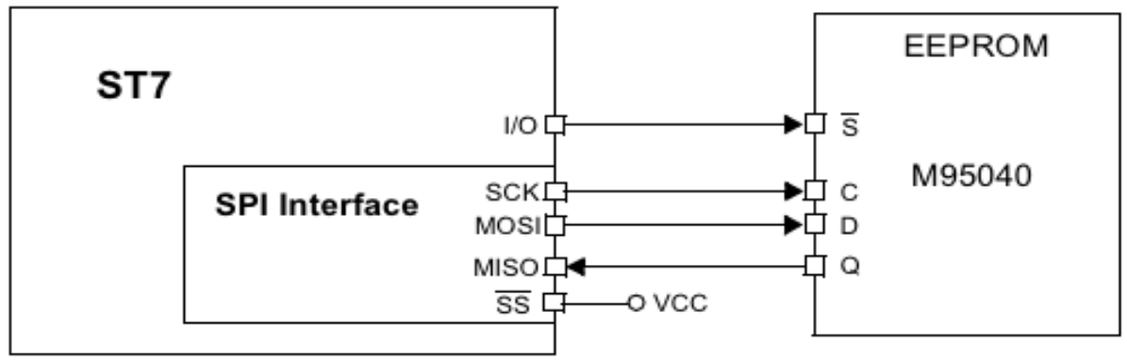
Electronically Erasable Programmable Read-Only Memory

- EEPROM are non-volatile memories used to store small amounts of data.
- EEPROMs are a forerunner to flash memory.
- Typically used to store system configuration/calibration data or long-term time-series data for a (field) data acquisition system.
- Interfacing to a Microcontroller is straightforward using one of several serial communication protocols.
- Storage/retrieval rates are slow (cf. internal memories) but EEPROMs are reliable and require little power.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Serial Memory Interfaces

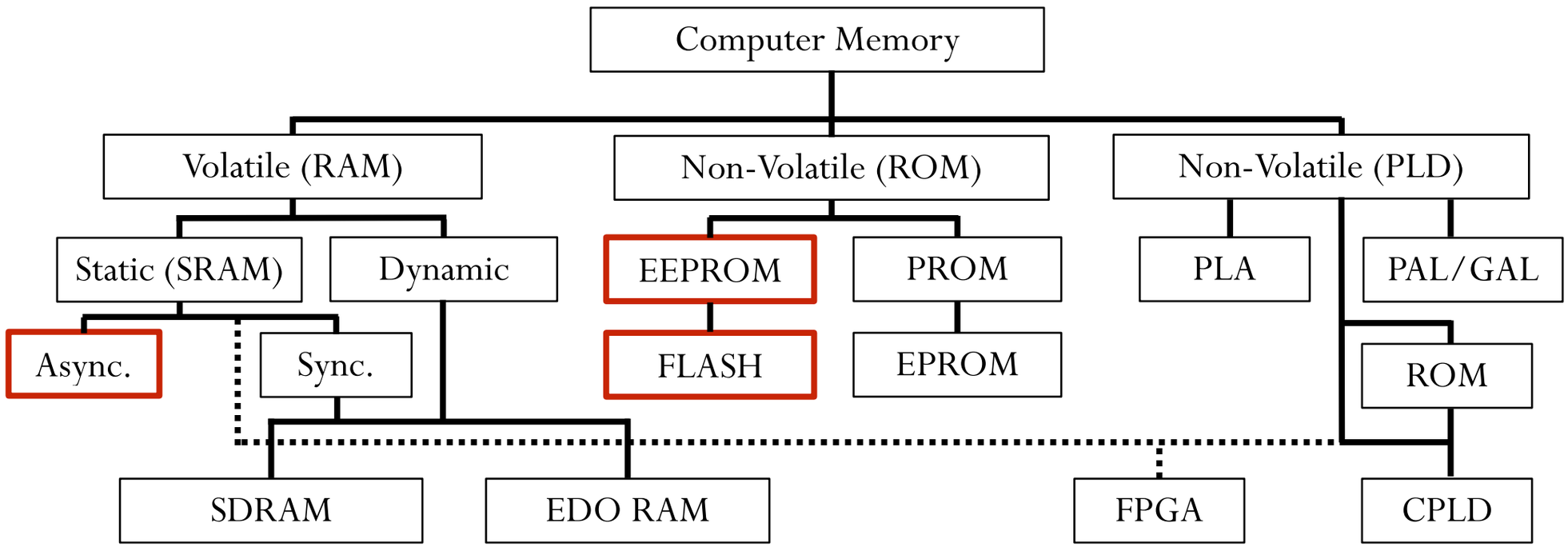
- Require synchronous (i.e. clocked) protocols.
- Advantages:
  - Fewer pins.
  - Lower crosstalk.
  - Simple interfacing, e.g. SPI, SSI, I²C.
- Disadvantages:
  - Speed? Though 10 Mbps is often enough for embedded applications.
  - Continuous mode minimises delay between packets, but frame errors can creep in.



#Advertisement: Serial interfaces are covered extensively in ENCE461!

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Taxonomy



Note 1: This is not an absolute list

Note 2: Primary acronyms are defined in the glossary

Prominent in modern embedded systems

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Flash

- Non-volatile
- Slow storage / retrieval
  - Faster than EEPROM, but can only erase by sector
- Modern MCUs store program in flash
- External flash commonly used for data storage

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Summary of MCU Memory Types

Type	Volatile?	Erase Size	Max Erase Cycles	Cost (per byte)	Speed
SRAM	Yes	Byte/Word	Unlimited	Expensive	Fast
DRAM	Yes	Byte/Word	Unlimited	Moderate	Moderate
EEPROM	No	Byte/Word	Limited	Expensive	Fast to read, slow to erase/write
Flash	No	Sector	Limited	Moderate	Fast to read, slow to erase/write

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory in Modern 32-bit MCUs

TI TM4C123x
ARM Cortex-M4 core with floating point
- CPU speed up to 120 MHz
- Up to 1-MB Flash
- 256-KB SRAM and 6-KB EEPROM
- 32-ch DMA

Atmel ATUC256L3U
32-bit AVR® mcu with fixed point DSP support, dual port SRAM, multi-bus
– 256 Kbytes internal high-speed Flash
  – Single-cycle Access up to 25 MHz
– 32 Kbytes internal high-speed SRAM
  – Single-cycle access at full speed
– 12 Peripheral DMA Channels

Microchip PIC32MZ2048ECM144
Less than $10 per unit in volume.
– 2MB Flash memory (plus an additional 160 KB of Boot Flash)
– 512KB SRAM memory
- 50 MHz External Bus Interface (EBI)
- 50 MHz Serial Quad Interface (SQI)

Toshiba TX19A/H1
Low-power 32-bit RISC, 100 MHz and floating-point
– 2MB Flash memory
– 10KB SRAM
– 8 DMA Channels

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory II

ENCE361: Design & Architecture: Lecture Block 2

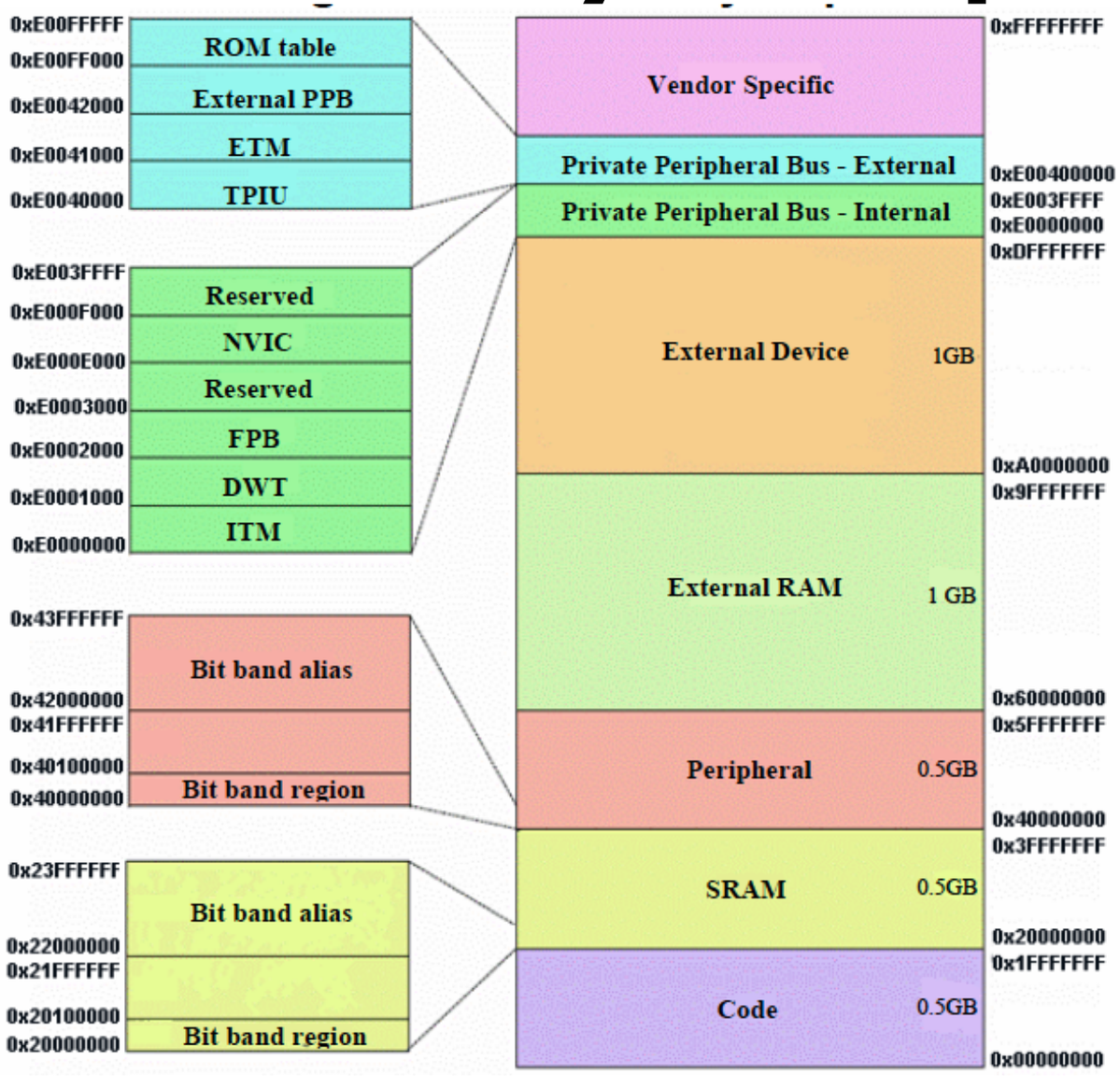
Outline

- Memory Organisation
  - Direct Memory Access
  - Endianness
  - Stack and Heap
- Memory Structures
  - Circular Buffers
  - Ring Buffers
  - Double Buffers

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Organisation

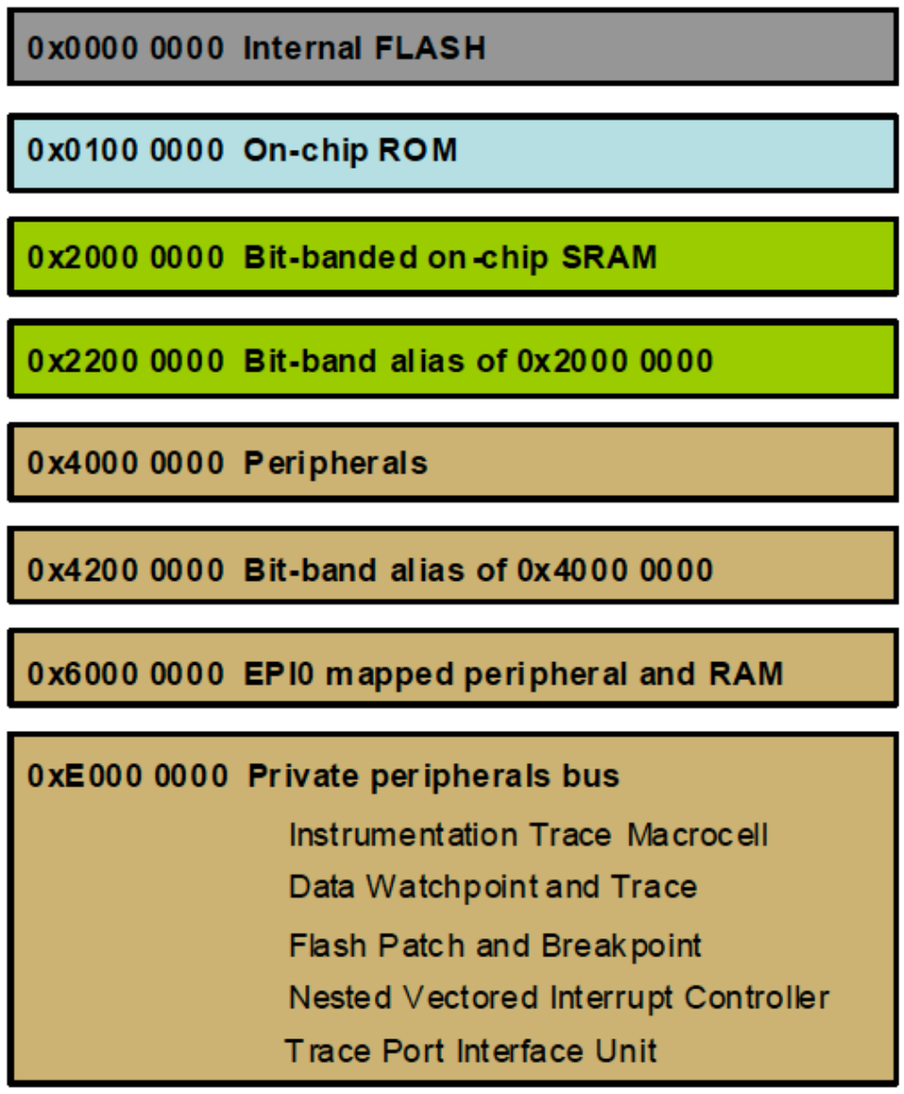
Cortex M-Series Memory Map



Ciaran Moore
ciaran.moore@canterbury.ac.nz

TM4C Memory Map

- **FLASH**
  - Up to 256 KB
  - **Single cycle up to 50MHz**
  - 1KB sector
  - Flash memory protection
  - Flash programming
- **SRAM**
  - 64KB
  - Up to 100MHz
  - Bit-band capability
  - Code execution
- **ROM**
  - Stellaris® Boot Loader
  - Stellaris® Peripheral Driver Library
  - AES cryptography tables
  - CRC error detection functionality



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Direct Memory Access

- DMA subsystems allow peripherals to read & write memory without the CPU being involved.
- They work by assigning control of the memory bus (address + data) alternatively to peripherals and the CPU.
- Often have multiple channels, each controlled by a set of registers (e.g. memory address, peripheral address, data size, configuration, DMA interrupt status, interrupt flag clear registers).
- Advantages of DMA:
  - More efficient use of interrupts
  - Increased data throughput
  - Reduced hardware costs as peripheral-specific FIFO buffers are not required

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Direct Memory Access

- Three modes of operation:
- Burst (block transfer) mode
  - An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system by the CPU, it transfers all data words in the data block before releasing control of the memory bus.
- Can leave CPU stalled for relatively long periods of time.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Direct Memory Access

- Three modes of operation:
- Burst (block transfer) mode
- Cycle stealing mode
  - Once DMA gains access to the system it transfers a single data word before handing control of the system bus back to the CPU.
  - CPU instructions and data transfers are effectively interleaved.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Direct Memory Access

- Three modes of operation:
- Burst (block transfer) mode
- Cycle stealing mode
- Transparent mode
  - DMA controller only transfers data when the CPU is performing operations that do not use the system buses.
  - Slowest for data transfer but most efficient in terms of system performance.
  - CPU never stops executing and DMA transfers happen for free.
  - Requires complex DMA hardware.

Endianness

- Big Endian: MSB (bit 31) goes in first memory location.

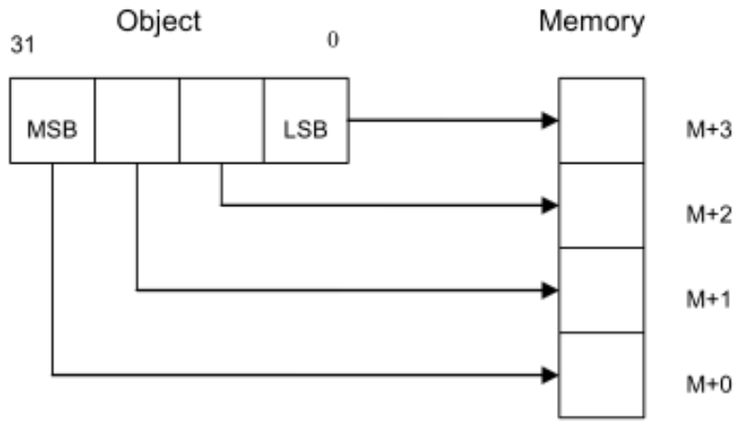


Figure 1, Memory layout of big-endian data object

- Little Endian: LSB (bit 0) goes in first memory location.

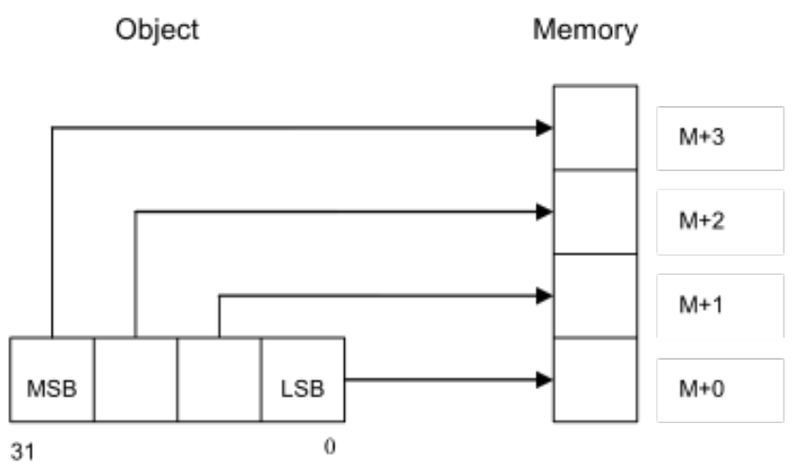


Figure 2, Memory layout of little-endian data object

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Endianness

- At reset, ARM memory access is Little Endian.
- Users can modify memory access to a Big Endian model.
- Historically, Motorola processors have supported Big Endian.
- Intel have promoted Little Endian.
- Endianness is only a problem when two or more processors attempt to access shared memory.

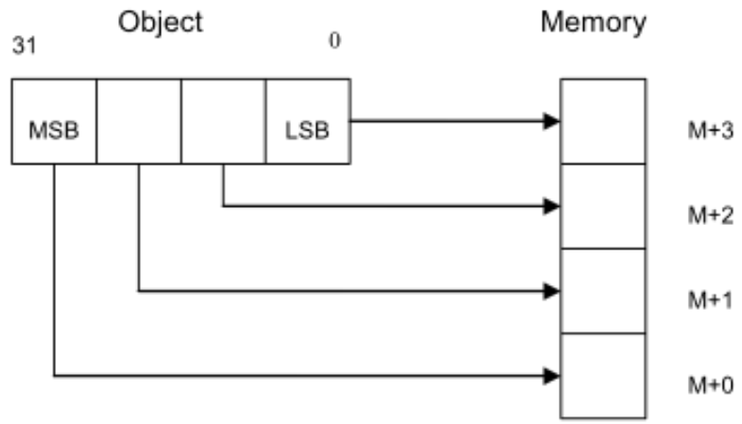


Figure 1, Memory layout of big-endian data object

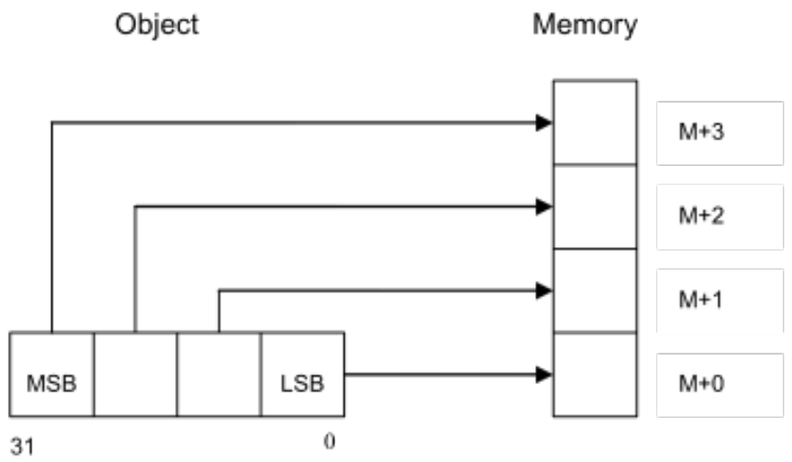
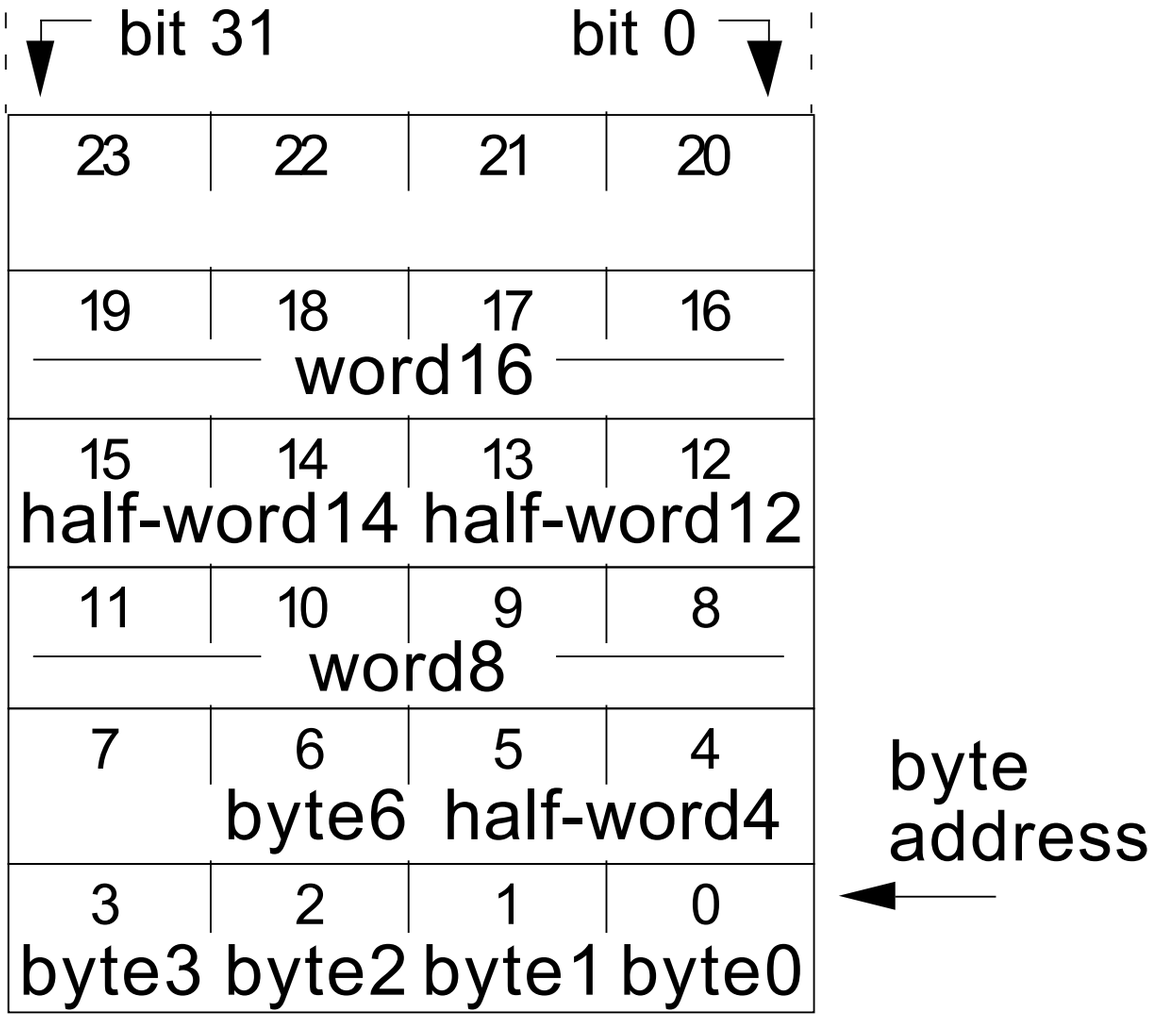


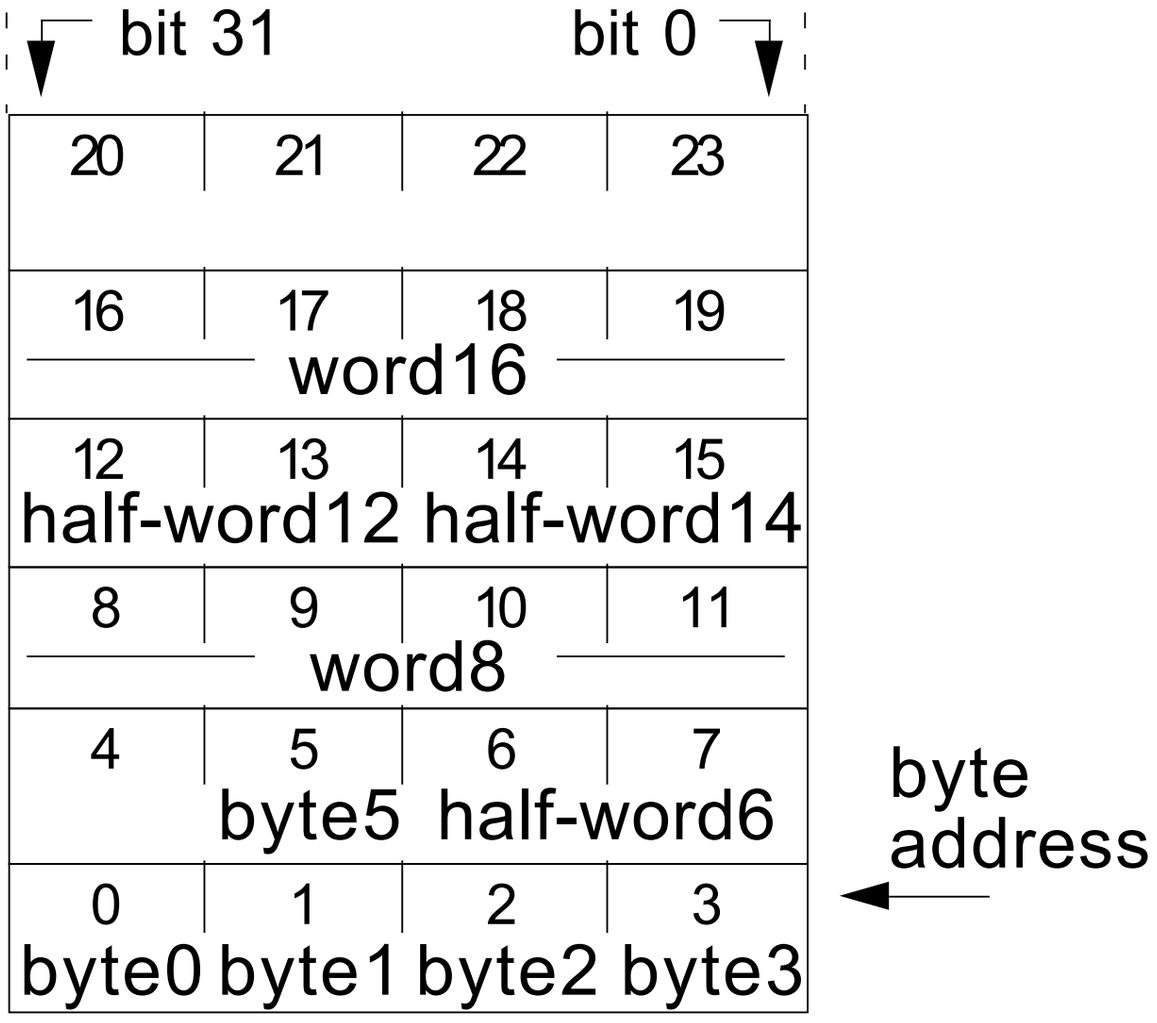
Figure 2, Memory layout of little-endian data object

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Endianness



(a) Little-endian memory organization



(b) Big-endian memory organization

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Alignment

- By default, ARM C compilers will store (or **pad**) data types on the following boundaries:
  - Bytes: any address
  - Half-words: only on even byte boundaries
  - Words: only on 4-byte boundaries
- Structures are also padded to the end of a word boundary.
- A programmer can enable more efficient memory usage (see bit *packing*), but this is **not** the default setting.

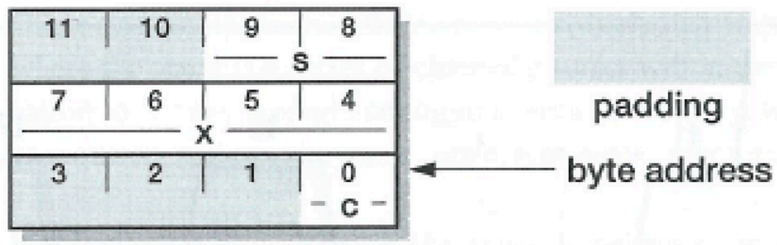


Figure 6.15 An example of normal struct memory allocation.

```
struct S2 {char c; short s; int x;} example2;
```

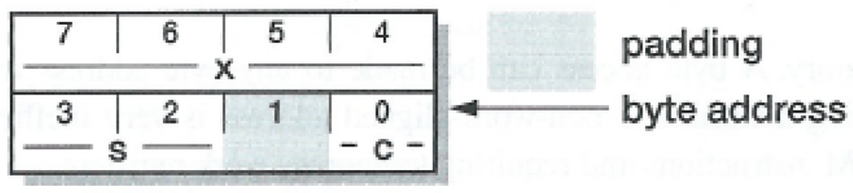


Figure 6.16 An example of more efficient struct memory allocation.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Stack and Heap

- Stack is a LIFO structure.
- Stack space is always in SRAM (normally at the top), as the stack contains data that changes often and needs to be read / written quickly.
- All local variables are stored on the stack.
- Interrupts and functions store CPU register contents and status information on the stack.
- Stack size can increase rapidly, especially for recursive functions.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Stack and Heap

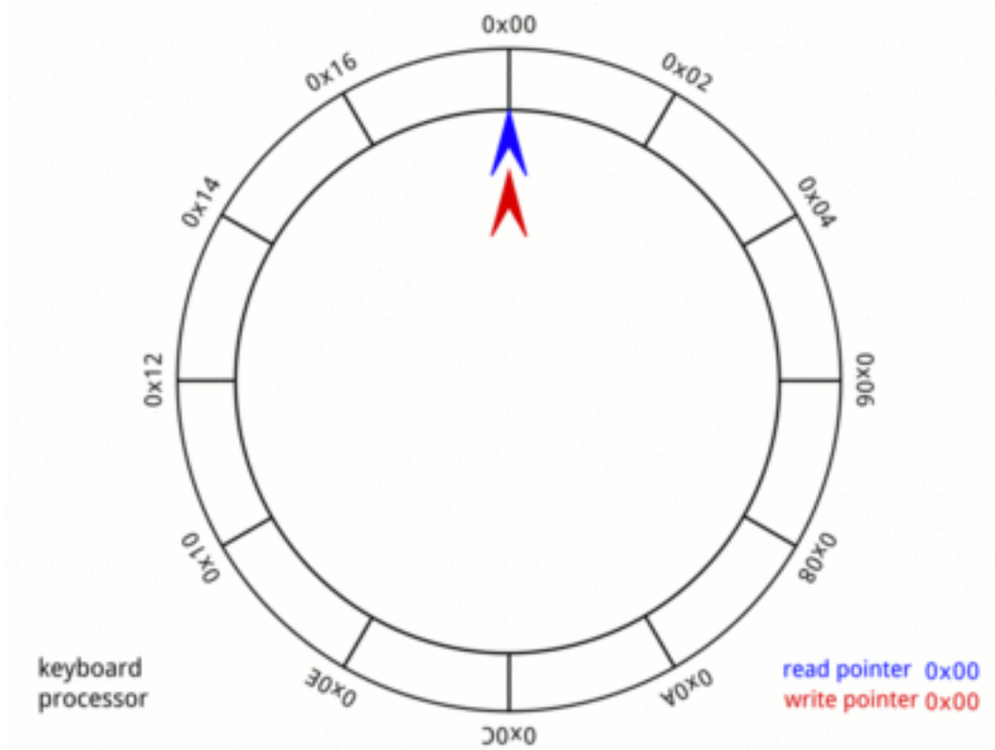
- Heap space is always in SRAM (normally at the bottom).
- Used for dynamically allocated (i.e. `malloc()`'d) structures.
- Stack and heap space needs to be reserved by the compiler before the program runs. (See Project Properties in CCS.)

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Memory Structures

Circular Buffers

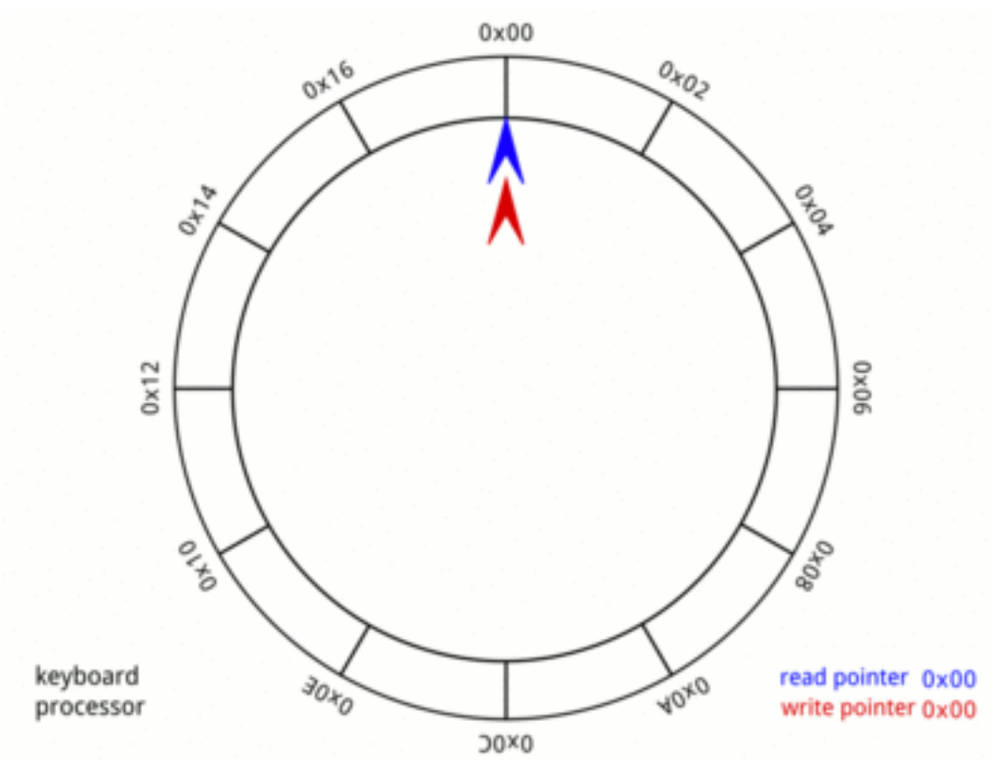
- A circular buffer of length *N* is a block of contiguous memory words addressed by pointers using a *modulo N addressing mode*.
- Both end addresses of the memory block are considered contiguous.
- Circular buffer constructs minimise operational overhead through data *abstraction*.
- Each circular buffer has two pointers:
  - A write (“head”) pointer and a read (“tail”) pointer.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Circular Buffers

- Characteristics of a circular buffer:
  - Instead of moving N data words in memory, just modify the pointers.
- When new data $x(n)$ arrives, the pointer is incremented and the new data is written in place of the oldest data.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Ring Buffers

- TivaWare has pre-defined memory structures (ring buffers) that can be used in place of your own structured circular buffer implementation.
- Located in `C:\ti\TivaWare_C_Series-2.1.4.178\utils`: `ringbuf.h` and `ringbuf.c`.
- Implement a FIFO structure with head and tail pointers.
- Queue length can vary from zero to full.
- Remember to **reserve enough stack space** when using these structures!

Ciaran Moore
ciaran.moore@canterbury.ac.nz

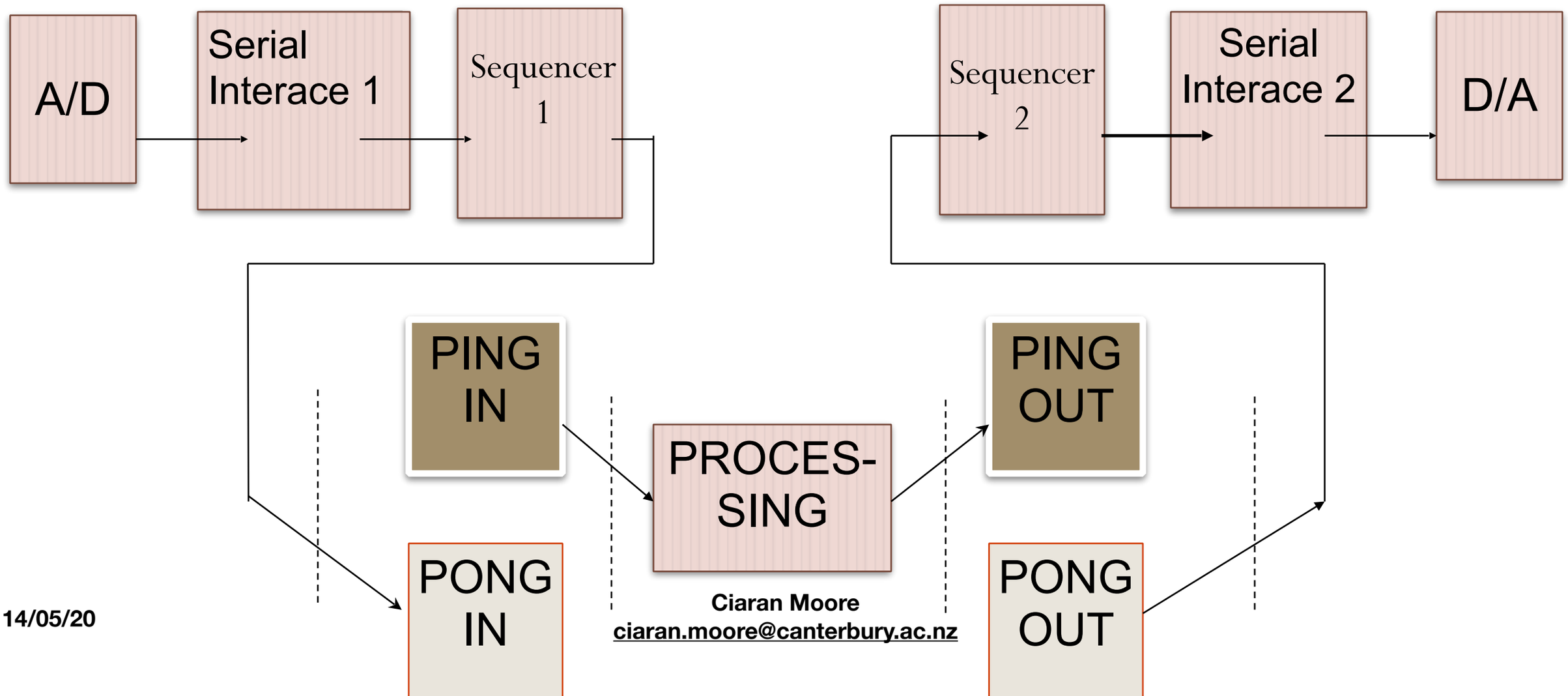
Double Buffering

- Two buffers are used: while one is being filled (written to) operations can be performed on data in the other.
- A flag is sued to signal that a buffer is full and can be operated on (read).
- Operations, such as reading and averaging a full buffer, are alternated every cycle.
- Useful for digital signal processing of audio and video.
- Prevents data corruption bugs.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Double Buffering

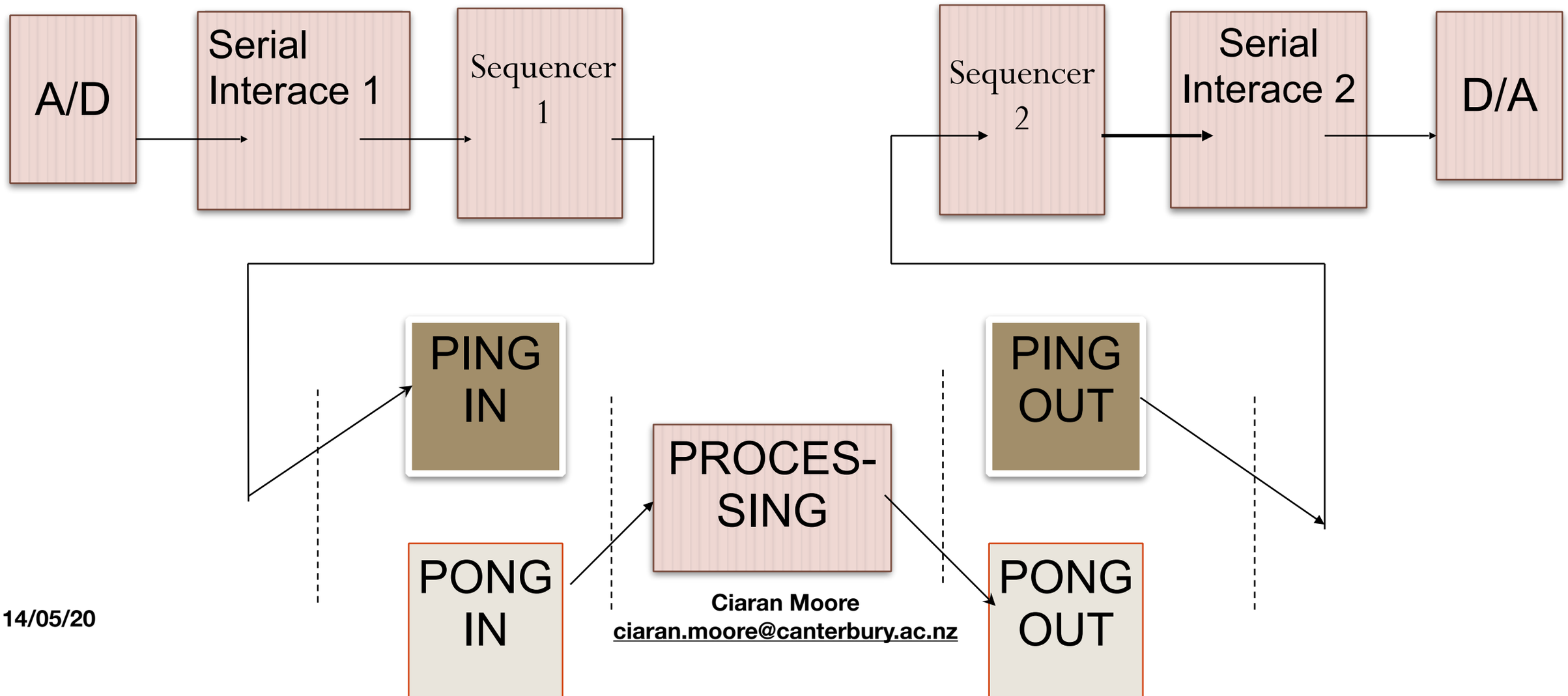
- Double circular buffering on four buffers: PING IN, PONG IN, PING OUT, PONG OUT
- Two separate memories are used and data sequencer channels are employed; one for the ADC and one for the DAC.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Double Buffering

- Double circular buffering on four buffers: PING IN, PONG IN, PING OUT, PONG OUT
- Two separate memories are used and data sequencer channels are employed; one for the ADC and one for the DAC.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Homework

ENCE361: Design & Architecture: Lecture Block 2

Homework

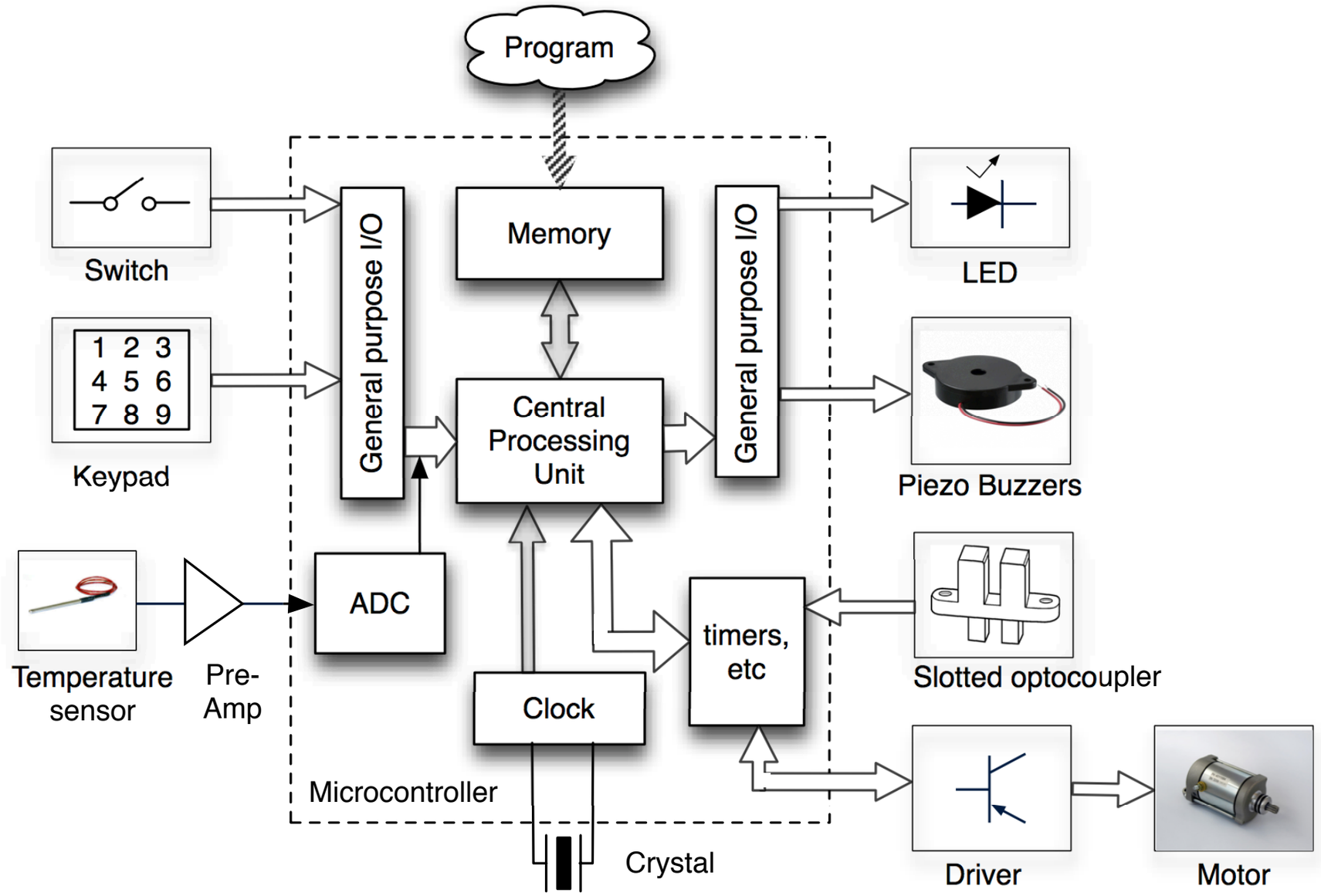
- On the figure on the next slide, highlight the components that are (a) sensors, (b) signal conditioning components, (c) actuators
- You're working with a sensor that outputs measurements over a 4-20 mA current loop. Assuming that the signal will be applied to an analog-to-digital converter (ADC) with a 0-5 V input range, what kind of signal conditioning would you apply to the sensor signal prior to the ADC input?

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Hardware Interfacing

- An embedded application typically comprises:

Sensors Microcontroller Actuators



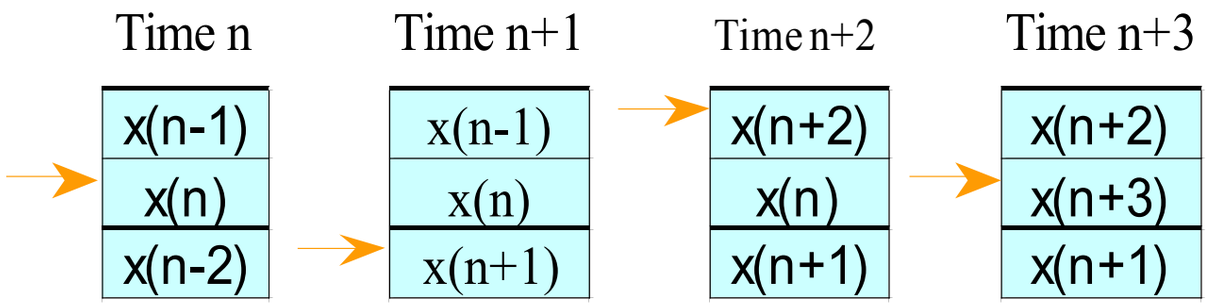
Homework

- Using four, 1K × 8-bit SRAM devices, redraw the schematic on slide 14 to create a 2K × 16-bit memory. In your answer show all data, address, and control lines.
- The TiVA TM4C123GH6PM Microcontroller has 256 kB of SRAM and 6 kB of EEPROM. Explain why the much smaller EEPROM is included when its contents could easily fit inside the SRAM.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Homework

- A circular buffer is updated at the time points shown below. In each case the oldest element is replaced by the newest element:



- Name the time-frame buffer (or buffers) where you would you expect to find the largest discontinuity in signal amplitude if the data stored are samples of a $3V_{p\text{-}p}$ sinusoidal waveform with a 1.5 V DC offset. The waveform starts at t=*0*; it is sampled at $\pi/2$ radians, and the buffer will be full at *t=n*, shown above, which occurs when the waveform is at π radians.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Homework

- From Furber, page 187, show how `example` below is allocated using a Little Endian memory model for the following memory structure:

```
struct S1 {char c; int x;};
struct S2 {
  char c2[5];
  uint16 c4[2];
  S1 s1[2];
} example;
```

- When implementing a circular buffer, is heap, stack or global memory preferred for implementation? Discuss, select and justify.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

References

- Furber, S., ARM system-on-chip architecture, 2nd Ed., Addison-Wesley, 2000.
- Atmel Corporation, AT91 ARM Thumb-based Microcontrollers Datasheet, Preliminary, November, 2006.
- Valvano, “Introduction to the ARM Cortex-M Microcontrollers”, 2017

References

- Mano M.M. & Kime C.R., Logic and Computer Design Fundamentals, Prentice-Hall, 2nd Ed. 2000.
- Morton. T.D. Embedded Microcontrollers. Prentice Hall. 2001.
- Lipovski G.J. Single and Multi-Chip Microcomputer Interfacing, Prentice-Hall, 1988.
- Wakerly J. F. Digital Design - Principles & Practices, 3rd Ed., Prentice Hall, 2000.
- Mano, M.M., Digital Design, Prentice Hall, 1984.

List of Acronyms

CPLD: Complex programmable logic device

DRAM: Dynamic RAM

EDO RAM: Extended data output RAM

EEPROM: Electronically erasable PROM

EPROM: Erasable PROM

FPGA: Field-programmable gate array

GAL: Generic array logic

PAL: Programmable array logic

PLA: Programmable logic array

PLD: Programmable logic device

PROM: Programmable ROM

RAM: Random access memory

ROM: Read-only memory

SDRAM: Synchronous DRAM

SRAM: Static RAM

Glossary

Access time: the time between a memory receiving a read signal and the data being available at the output.

Address: an unsigned integer that identifies the location of a word in memory.

Bit: the smallest quantum of data: A zero or one.

Byte: typically eight bits.

Cell: A device or circuit used to store a single bit of information.

EEPROM: A non-volatile memory that can be erased and programmed down to the bit level.

Flash: A (very) low-cost type of EEPROM that is sector erasable and programmable.

Read operation: copies information from memory to another location. Read operations are non-destructive.

Sector: Typically 4096 bytes (4 KiB).

Volatile: memory contents is lost if power is removed. cf. non-volatile memory.

Word: The contents stored at a memory address. Typically two bytes.

Write: stores new information in memory. Overwrites old information if required.

Ciaran Moore
ciaran.moore@canterbury.ac.nz