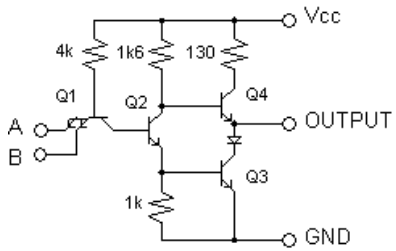
Lecture 5
Connecting peripherals

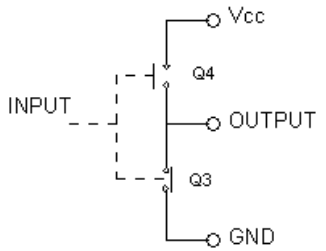
EE579
Advanced Microcontroller Applications
Dr James Irvine, EEE

TTL Output

- Example of basic circuit from a NAND gate

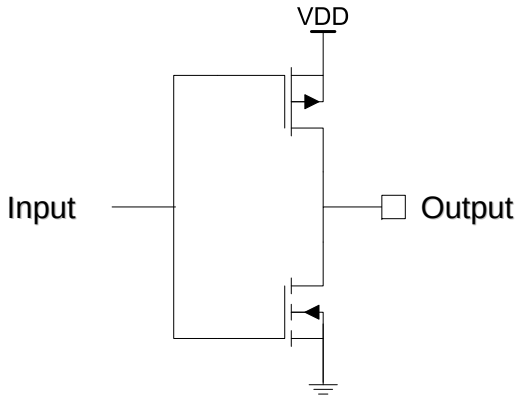


7400 NAND GATE



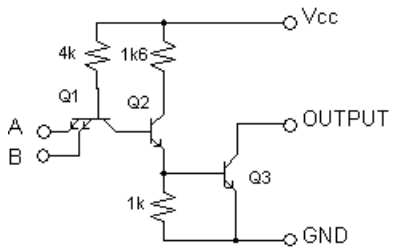
TOTEM-POLE OUTPUT

CMOS Output Circuit

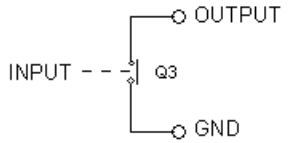


Open Collector Output

- TTL example, but name and principle is the same for other technologies

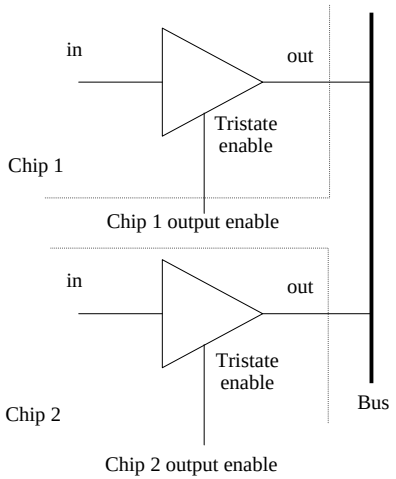


7403 NAND GATE



OPEN-COLLECTOR OUTPUT

Bus Connections

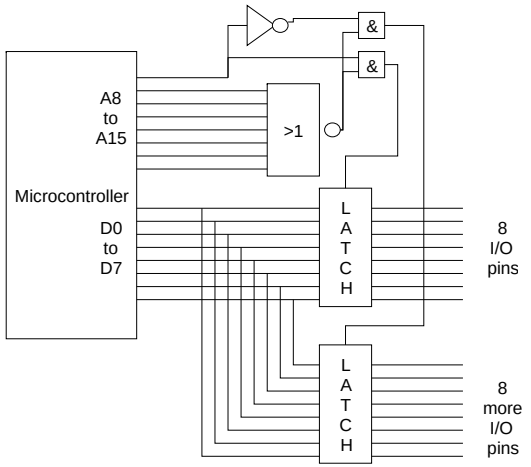


In	tristate	out
0	0	high impedance (Z)
1	0	high impedance (Z)
0	1	0
1	1	1

Peripheral Addressing

- I/O pins are a scarce commodity on a microcontroller
- Several devices can be connected to the microcontroller by using the address bus
- High order addresses can be decoded to select individual devices, while low order addresses can be used to address individual registers within the devices

Peripheral Addressing

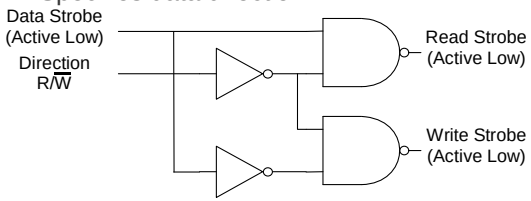


Peripheral Addressing - Basic Signals

- Address to specify peripheral ⇒ Chip Select
- Address to specify register on chip
- Reading or writing then requires
- Device enable
- Direction

Basic Signals - Variations on a Theme

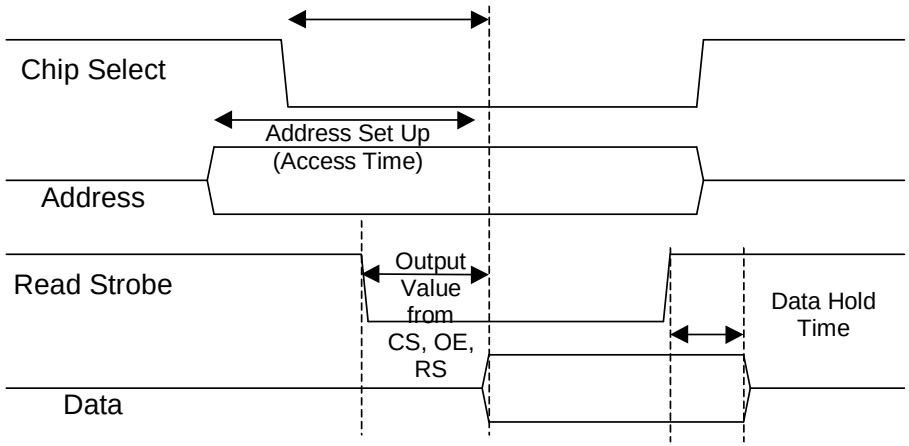
- Intel Devices
  - Read Strobe : Enables device and specifies read
  - Write Strobe : Enables device and specifies write
- Motorola, Arizona Microchip (PIC)
  - Data Strobe : Enables device
  - R/W : Specifies data direction



Access Procedure - Read

- Assert the address and stabilise it
- Select the chip
- Assert the data strobe
- Wait for the data to stabilise
- Read it
- Release strobe, and deselect chip
- Note - read usually done on release of strobe

Access Times - Read



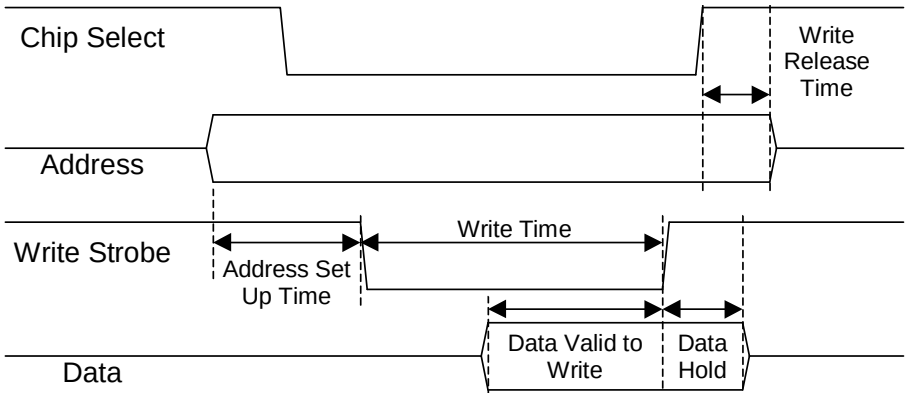
Access Procedure - Read

- Timings are relatively simple - minimum time for the output data to be stable defined from assertion of control signals
- Time from address stable likely to be substantially longer than for control lines, but remember decoding logic
- For busses with separate direction, the direction control may have a minimum set up time prior to chip selection to avoid false writes

Access Procedure - Write

- Assert the address and stabilise it
- Select the chip
- Assert the data
- Assert the data strobe
- Release strobe, and deselect chip
- For compatibility with read operations, the data strobe can be asserted before the data is stable, as long as the data is stable before the end of the strobe

Access Times - Write



Access Time - Write

- Write Accesses are more complex than reads
- The address must be stable before write is asserted to avoid writing to the wrong location
- Data is latched to release of the write strobe - it must be sable for a period before, and sometimes held after
- The address must be held after the write to avoid any false address being written to

Types of RAM

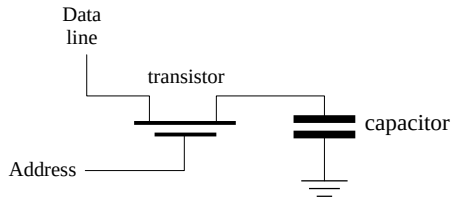
- Volitile
  - Static RAM
  - Dynamic RAM
- Non volitile
  - Battery backed up RAM
  - Flash
  - EEPROM
- RAM may be ‘dual port’
  - allows access by two devices
  - may be used for inter-device communication
  - requires careful synchronisation

Types of RAM

- SRAM uses flip flops, retains data as long as power is available
- DRAM uses a capacitor, giving the advantages of
  - higher densities (one transistor per memory element)
  - lower cost
- BUT
  - charge on the capacitor decays, so DRAM needs refreshing
- Embedded systems usually use SRAM - DRAM is only used on large systems

DRAM

- One transistor, one capacitor
- Transistor connects data line to capacitor
- Writing - capacity charges to level of data line
- Reading - capacity asserts its voltage on the data line

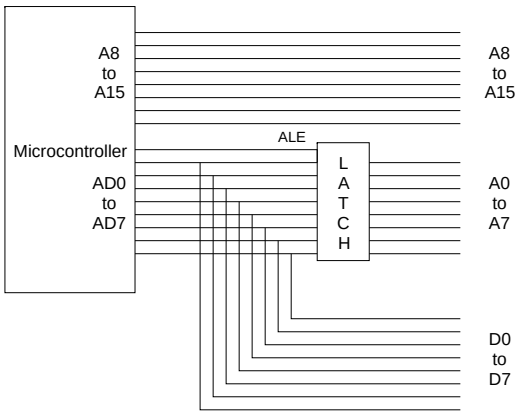


DRAM

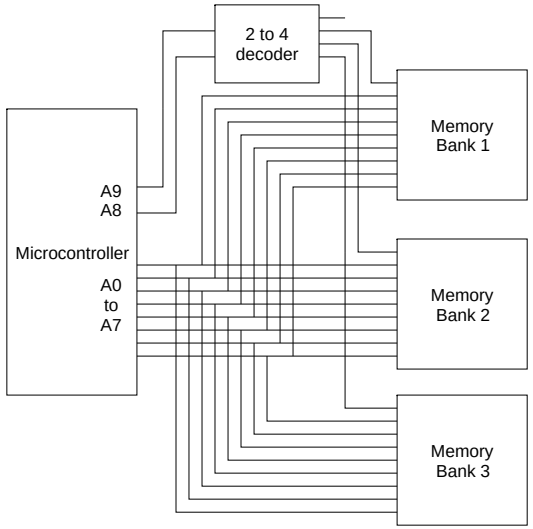
- DRAM have two addresses - row address and column address
- To refresh, assert the row address - internal circuitry on the memory chip will refresh all elements in that row
- To access, assert row address and strobe (RAS), then change address to column address and assert column strobe (CAS). For write, assert write and data prior to CAS.
- Remember to refresh if automatic circuitry not available

Saving I/O Pins

- Address Latching
  - Great saving in IO
  - But, reduces effective bus speed
  - From the days of DIP packages, so now obsolete
  - Still to be found on the 8051 in uP mode
- Bank Switching



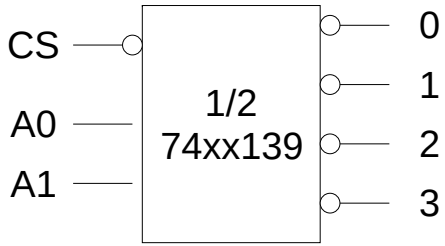
Address Decoding



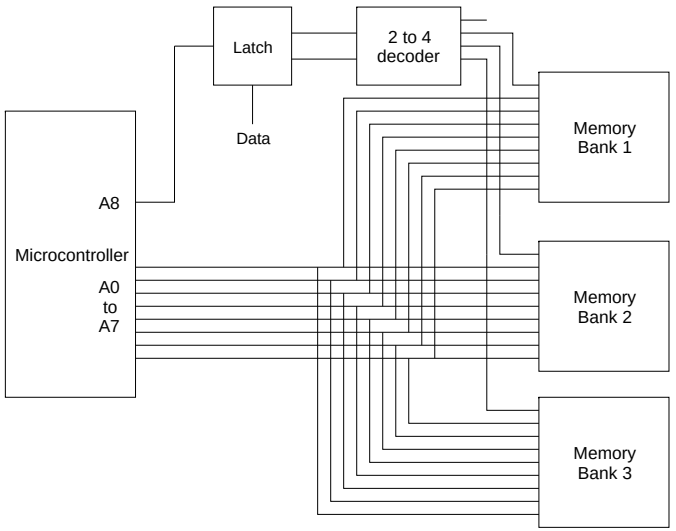
Decoders

- Prepackaged (normally) $\log_2(n)$ to n circuits
- 74xx139 is a dual (i.e. there are two in the package) 2 to 4 decoder chip
- 74xx138 is a 3 to 8 decoder chip
- 74xx159 is a 4 to 16 decoder chip
- Decoders will have additional enable inputs

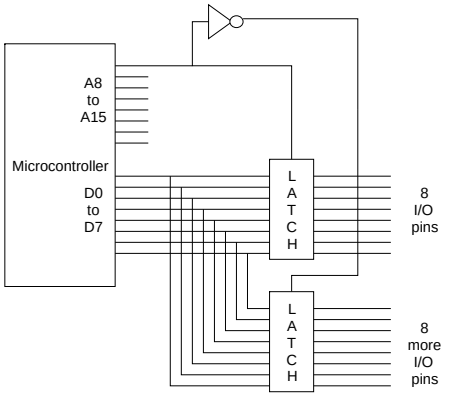
74xx139



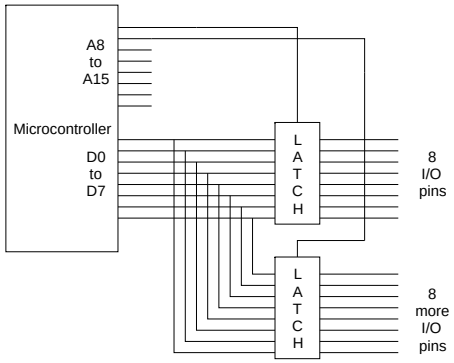
Bank Switching



Partial Addressing



Less hardware



Even less hardware, but be careful - both latches could be addressed at once

Bus Loading

- BEWARE - bus loads soon add up if you are connecting many peripherals
- Timing values in the data sheets will be for a specified load, and will extend significantly if the bus is overloaded
- Buffering the bus can sometimes speed things up, since the propagation delay is less than the delay caused by overloading the chip's outputs

Summary

- Study the data sheets carefully
- Don't get minimums and maximums mixed up
- Watch the delay through address decoding
- Watch the load, especially for control signals like R/W
- Timing faults are best avoided - they can be very, very difficult to track down

LCD Display

- Connected to P2
- Register select pin P2_0
- Enable pin P2_1
- Data (4 bits) P2_4, P2_5, P2_6, P2_7
- Busy bit not connected (!)
- **Watch the timings**

LCD_Display.h

```
void InitialiseDisplay( void );
void DisplayString(unsigned char position, char
   *string);
void LCD_write(unsigned char data_or_ctrl,
   unsigned char value);
void LCD_nibble_write(unsigned char
   data_or_ctrl, unsigned char value);
void DisplayDelay(unsigned long int units);
```