09 - Low Power Design

CEG 4330/6330 - Microprocessor-Based Embedded Systems

Max Gilson

Low Power Design Considerations

- Sometimes embedded systems run off of a battery
- Sometimes many embedded systems are used in tandem
- It is important to consider the power limitations of your system when you design it
  - How long should your phone/drone/device run with a battery?
- Often you must choose a trade off between performance and power consumption

Methodologies for Reducing Power Consumption

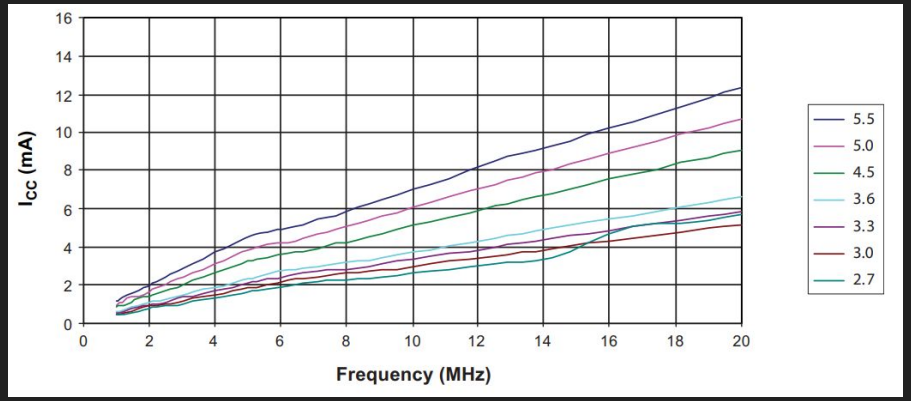
- Static
  - Use slower communication rates
  - Use slower clock frequencies
  - Shut down unused components
- Dynamic
  - Voltage and frequency scaling
    - For CMOS circuits, power dissipation is proportional to $C \times V^2 \times F$
  - Shut down unused components

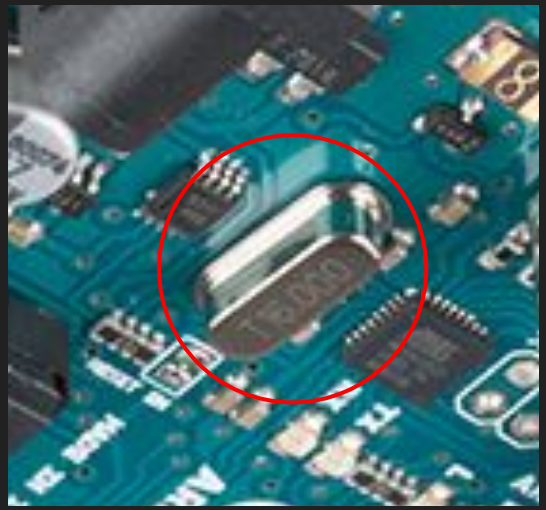
Static Methods

- Use slower communication rates
  - An ambient temperature sensor usually does not need to be read 1000 times per second, reading it less often saves power
  - A high baud rate is not always required, use a slower baud rate to save power

Static Methods (cont.)

- Use slower clock frequencies
  - Replace the ATMega328p's 16MHz clock with 8MHz or even 1Mhz!
    - Nearly 80% power savings





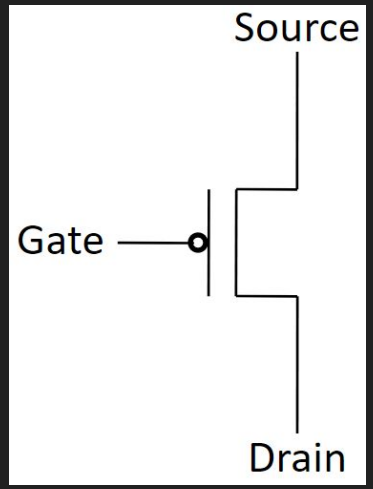
Static Methods (cont.)

- Shut down unused components
  - When a device is not in use shut it down or put it into low power mode
  - If you only need to record a temperature once per minute, shut it down in between reads
  - If possible, put the microcontroller/microprocessor into low power mode as well

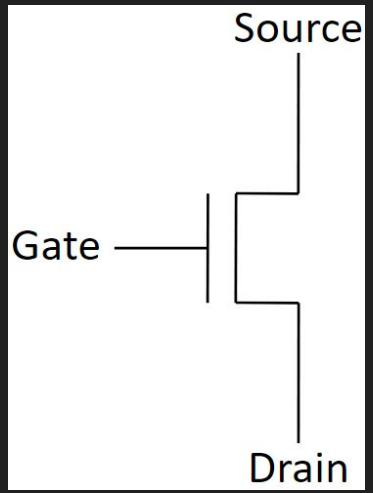
CMOS

- Complementary metal–oxide–semiconductor (CMOS)
  - P-type: gate = 0 volts ⇨ closed switch (current flows)
  - N-type: gate = 2.9 volts ⇨ closed switch

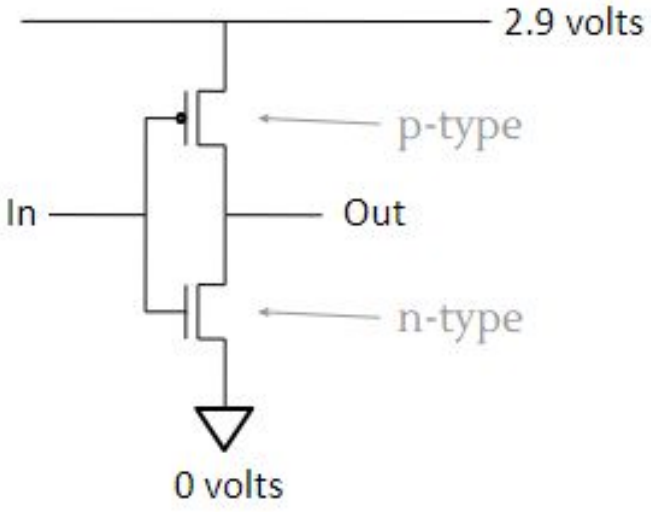
P-type:



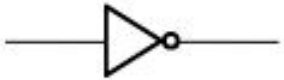
N-type:



Combining Transistors: Logic Gates



A CMOS inverter



Voltage table

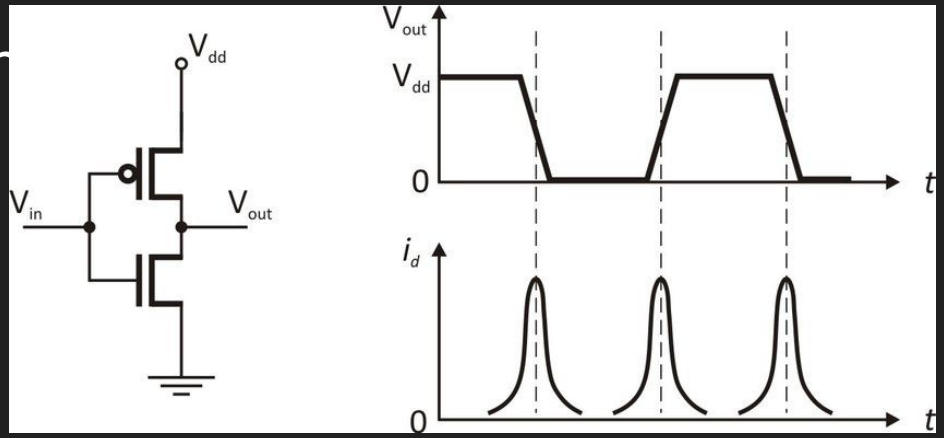
In	Out
0 v	2.9 v
2.9 v	0 v

Inverter truth table

In	Out
0	1
1	0

Dissipation Due to Direct-Path Currents

- A short circuit occurs when both p-type and n-type transistors change simultaneously
  - Transistors are not ideal switches!
- Static power consumption is 0W



$$P_{dp-avg} = t_{sc} \cdot V_{DD} \cdot I_{peak} \cdot f$$

$$I_{peak} : \ peak\, current$$

$$t_{sc} : \ short\, circuit\, duration$$

$$f : \ switching\, frequency$$

$$V_{DD} : \ supply\, voltage$$

Power Saving in CMOS - Voltage

- Lower the VDD to save power
  - Noise margin is a constraint you must accept
  - If too low of a voltage is used, noise compromises the logic signals
  - If too high of a voltage is used, logic consumes more power
- Some devices allow for a wide range of stable input voltages
  - It may be risky to power close to the minimum voltage

Power Saving in CMOS - Frequency

- Lower the frequency to save power
  - More frequency usually means more performance
  - If your design does not require high performance, lower the clock frequency
- Some microprocessors and microcontrollers allow dynamic frequency scaling
  - Change the frequency at runtime when the device requires more/less power

ATmega328p Low Power Mode

- SMCR - Sleep Mode Control Register
  - Determines 1 out of 6 low-power modes
  - Each mode has restrictions/limitations

Table 9-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources							
	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other/O	Software BOD Disable
Idle			X	X	X	X	X(2)	X	X	X	X	X	X	X	
ADC noise Reduction				X	X	X	X(2)	X(3)	X	X(2)	X	X	X		
Power-down								X(3)	X				X		X
Power-save					X		X(2)	X(3)	X	X			X		X
Standby(1)						X		X(3)	X				X		X
Extended Standby					X(2)	X	X(2)	X(3)	X	X			X		X

Notes: 1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

- **Bits 3..1 – SM2..0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the five available sleep modes as shown in Table 9-2 on page 38.

Table 9-2. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC noise reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby(1)
1	1	1	External standby(1)

Note: 1. Standby mode is only recommended for use with external crystals or resonators.

- **Bit 0 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the sleep enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

Manually Disabling Timers

- Power Reduction Register - PRR
- Allows you to disable specific features
- Mostly used for idle sleep mode

Raspberry Pi 4B Low Power Options

- The Raspberry Pi is a more complex device than Arduino Uno
- Ways to save power consumption:
  - Disable USB (~0.50W)
  - Disable HDMI (display output) (~0.15W)
  - Disable communications (WiFi and Bluetooth) (~0.40W)
  - Disable Onboard LEDs (~0.05W)
  - Reduce clock frequency
    - Clock frequency is changed by manipulating an external supply voltage (sometimes called VDD-CPU)
- Possible to save about ~40% idle power consumption

CPU Frequency Governors

- Operating systems are capable of managing the CPU frequency on the fly for many microprocessors
- You can select a governor to optimize for performance or power savings
  - Performance Governor - sets the CPU frequency statically to the highest frequency limit
  - Power Savings Governor - sets the CPU frequency statically to the lowest frequency limit
  - Ondemand Governor - sets the CPU frequency based on CPU load

Hibernation

- Some operating systems allow for hibernation (Raspberry Pi 4B does not)
- Hibernation shuts down RAM and puts computer into a deep sleep
- Contents of RAM must be saved to external storage before hibernation
- When RAM is shut down, all contents are lost until reloaded from external storage