Chapter 4

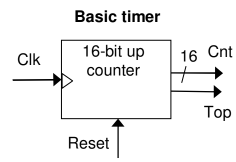
Standard Single-purpose processors

Introduction

- Single-purpose processors
  - Performs specific computation task
  - Custom single-purpose processors
    - Designed by us for a unique task
  - *Standard* single-purpose processors
    - "Off-the-shelf" -- pre-designed for a common task
    - a.k.a., peripherals
    - serial transmission
    - analog/digital conversions

Timers, counters, watchdog timers

- Timer: measures time intervals
  - To generate timed output events
    - e.g., hold traffic light green for 10 s
  - To measure input events
    - e.g., measure a car's speed
- Based on counting clock pulses
    - E.g., let Clk period be 10 ns
    - And we count 20,000 Clk pulses
    - Then 200 microseconds have passed
    - 16-bit counter would count up to 65,535*10 ns = 655.35 microsec., resolution = 10 ns
    - Top: indicates top count reached, wrap-around

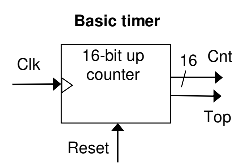


Timers, counters, watchdog timers

- Timer: measures time intervals
  - To generate timed output events
    - e.g., hold traffic light green for 10 s
  - To measure input events
    - e.g., measure a car's speed
- Based on counting clock pulses
    - E.g., let Clk period be 10 ns
    - And we count 20,000 Clk pulses
    - Then 200 microseconds have passed
    - 16-bit counter would count up to 65,535*10 ns = 655.35 microsec., resolution = 10 ns
    - Top: indicates top count reached, wrap-around

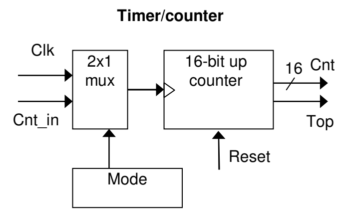
Resolution

Range



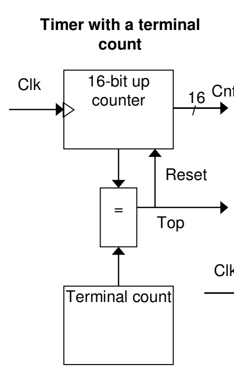
Counters

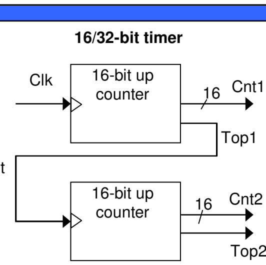
- Counter: like a timer, but counts pulses on a general input signal rather than clock
  - e.g., count cars passing over a sensor
  - Can often configure device as either a timer or counter

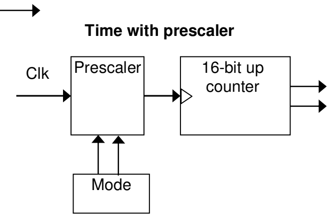


Counter/Timer structures

- Interval timer
  - Indicates when desired time interval has passed
  - We set terminal count to desired interval
    - *Number of clock cycles = Desired time interval / Clock period*
- Cascaded counters
- Prescaler
  - Divides clock
  - Increases range, decreases resolution

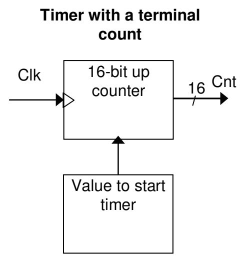






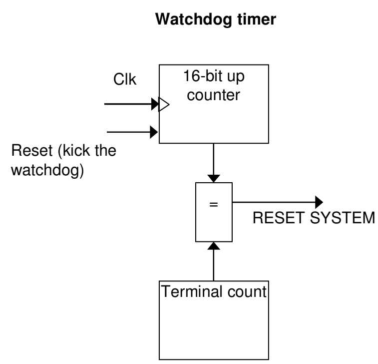
Counter/Timer structures

- Timer with ability to be initialized.



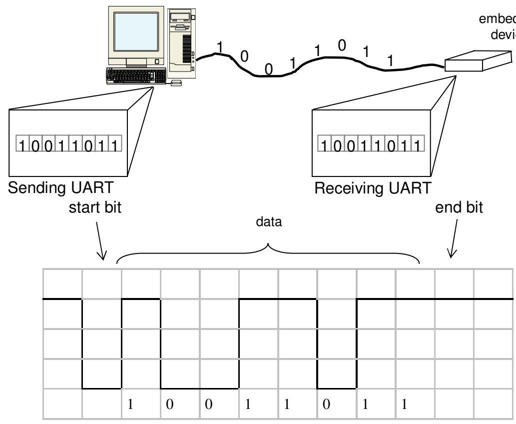
Watchdog timer

A watchdog timer is similar to a counter with a terminal count, but it is used to "watch" over the system, so that if the watchdog is not reset frequently enough it will indicate that something it wrong.



Serial Transmission Using UARTs

- UART: Universal Asynchronous Receiver Transmitter
  - Takes parallel data and transmits serially
  - Receives serial data and converts to parallel
- Parity: extra bit for simple error checking
- Start bit, stop bit
- Baud rate
  - signal changes per second



Parity Bit

The parity bit is for *simple* error checking.

If your 8 bit data is:

00110101

than the encoded value of 9 bit ODD parity is:

001101011

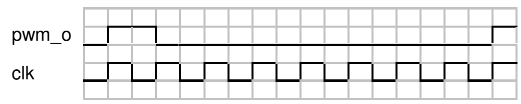
↖ Parity bit

Your receiver will have to know that the transmitter is encoding data in 9 bit odd parity. Then it will read in 9 bit, check to make sure there are an odd number of one's and then save the data. If there are an even number of one's it will know there was an error, and either toss the data, or request a resend.

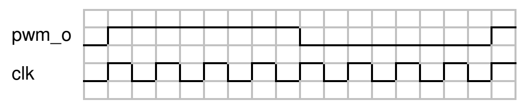
Other more complex error checking methods include CRC, and 8B/10B

Pulse width modulator

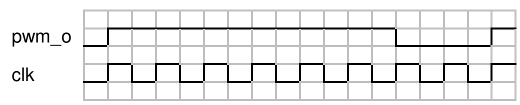
- Generates pulses with specific high/low times
- Duty cycle: % time high
  - Square wave: 50% duty cycle
- Common use: control average voltage to electric device
  - Simpler than DC-DC converter or digital-analog converter
  - DC motor speed, dimmer lights



1/8 = 12.5% duty cycle: 5V * 0.125 = 0.625V

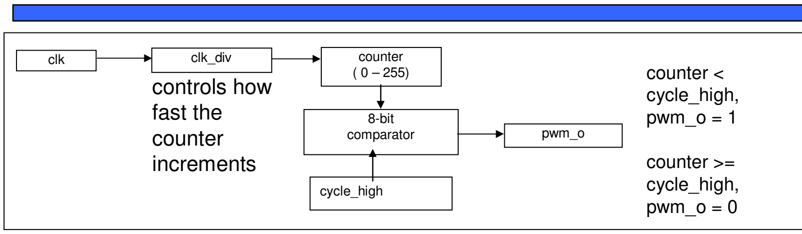


50% duty cycle – average pwm_o is 2.5V.



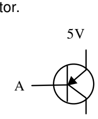
75% duty cycle – average pwm_o is 3.75V.

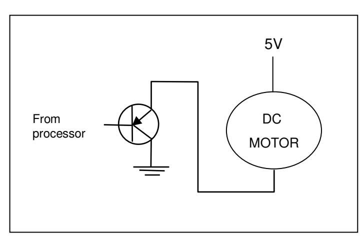
Controlling a DC motor with a PWM



Internal Structure of PWM

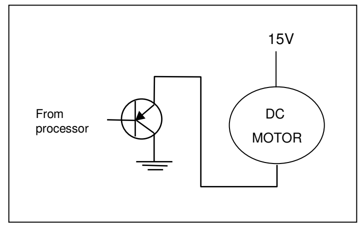
The PWM alone cannot drive the DC motor, a possible way to implement a driver is shown below using an MJE3055T NPN transistor.





Example of PWM control of a motor

If a motor has a speed of 10200 RPM when 15V is applied and you are controlling it with a PWM that has 8 bit resolution, what values do you need to write to the PWM to get closest to the following speeds: a) **10,000** RPM, b) **100** RPM, and c) **5,100** RPM?



Example of PWM control of a motor

For an 8-bit PWM, if you want to set a motor to run at 10,000 RPM, and you know that full power will result in 10,200 RPM:

10000 / 10200 = 98.04%

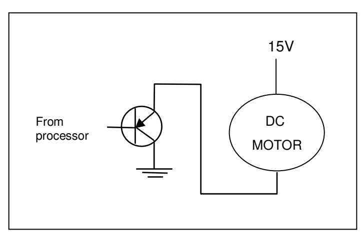
255 * .9804 = 250.002

Closest value = 250

= 0xFA

for 100: 2.5 (round up or down)

for 5100: 127.5



Midterm Review

- Chapter 1:
  - Definitions:
    - Design Metrics (NRE, Unit Cost, Latency, Throughput, and many more)
    - Design Productivity Gap
    - Mythical Man-Month
  - Processor Technology: General, App. Specific, Single
    - Know the advantages and disadvantages of each
  - IC technology: Full Custom, Semi-Custom, PLD
  - Moore's Law

Midterm Review

- Chapter 2:
  - Custom Single Purpose Processors
  - Know Combinational Logic vs. Sequential Logic and timing
  - Know what transistors do, and how to make inverters, nands and nors
  - Remember Karnaugh Maps
  - **Know every step in creating a single purpose processor from an algorithm**
  - Know how to optimize SPP's at every level

Midterm Review

- Chapter 3:
  - General Purpose Processors
  - Know the basic parts of a processor
    - PC, IR, Controller, Datapath (ALU, Registers), main memory
  - Know how data moves in a GPP
  - Know some performance improvers (pipelining, multiple ALU's, etc.)
  - Know Harvard vs. Princeton memory arch.
  - Know what a cache is
  - Understand the different addressing modes
  - Know how to create a program in assembly
  - Know the development environment and testing terms

Midterm Review

- Chapter 4:
  - Understand counters and timers and the terms associated with them.
  - Understand Parity Bits.
  - Know how to use and set a pulse width modulator