Carnegie Mellon University Qatar

15-348: Embedded Systems Lecture 15

Fall 2022

Ryan Riley
(based on original slides by Saquib Razak)

1

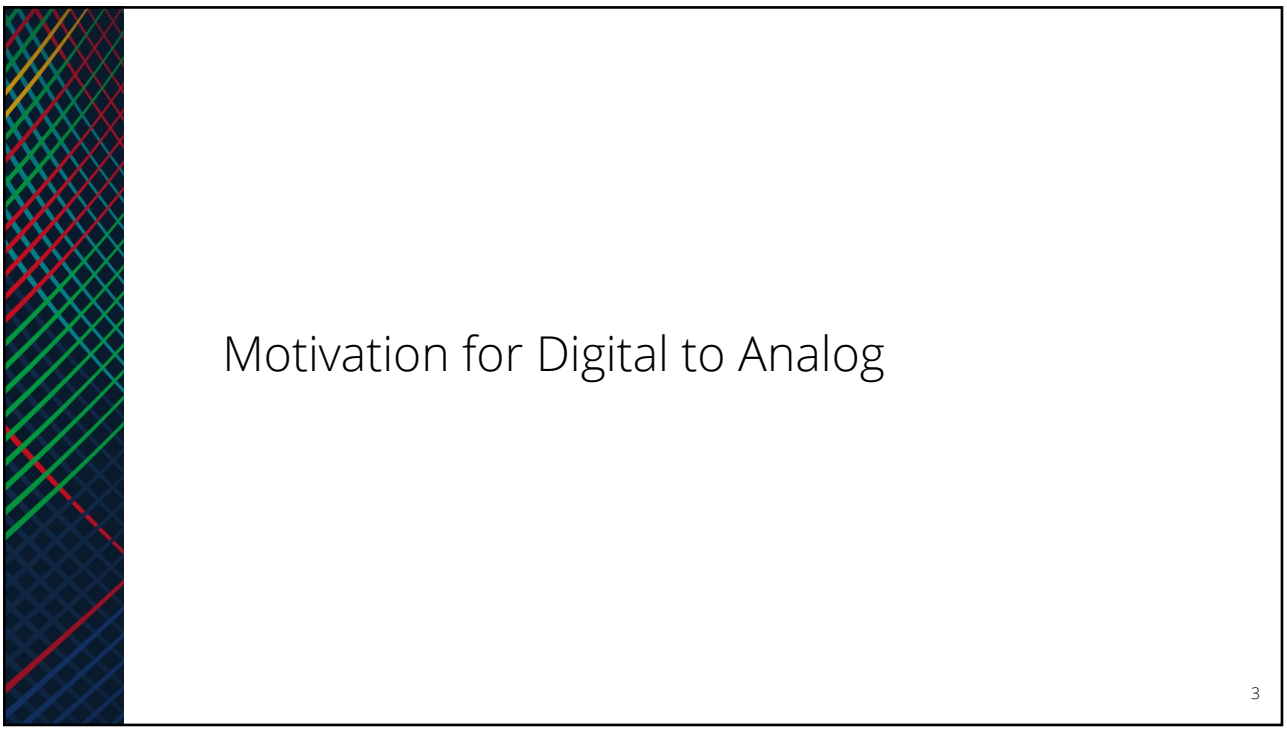
1

Where are we?

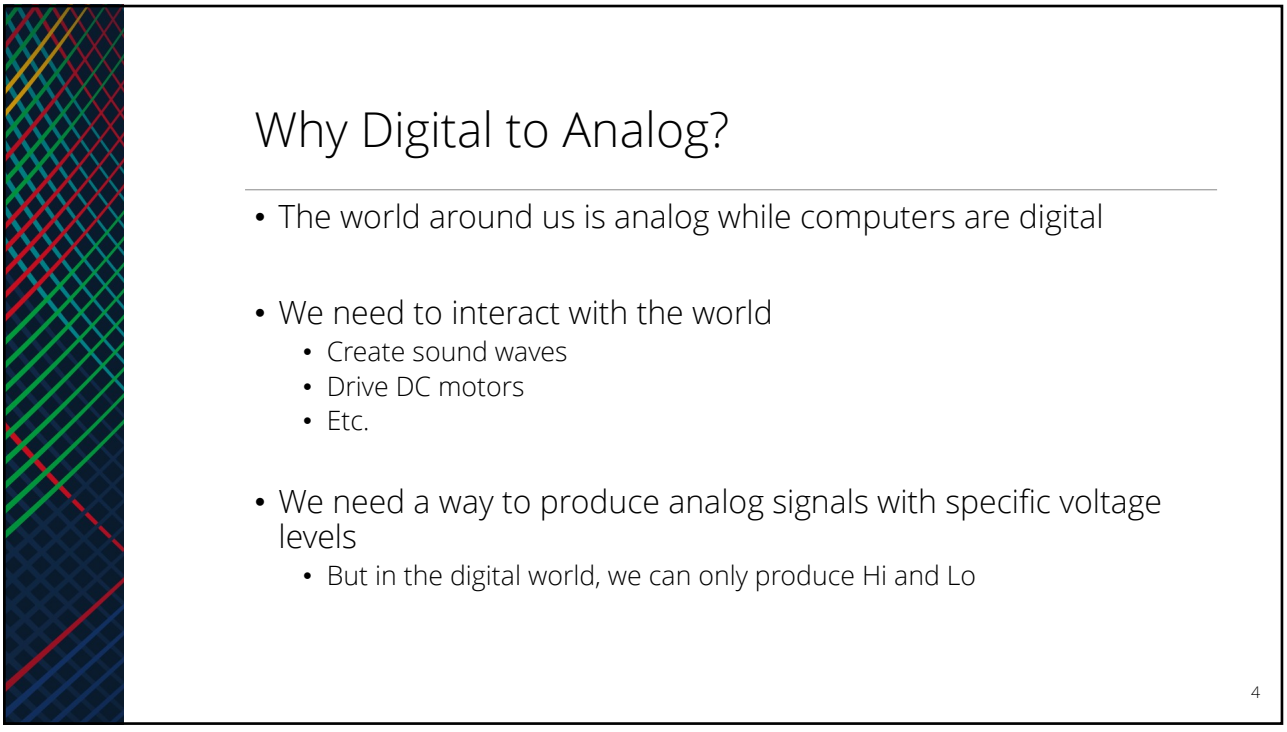
- Recently
  - Analog to Digital
- Today
  - Digital to Analog
- Reading
  - Pages 309-312, 411-423 in the book
  - Interesting in other DAC designs? Try… https://www.analog.com/media/en/training-seminars/design-handbooks/Basic-Linear-Design/Chapter6.pdf

2

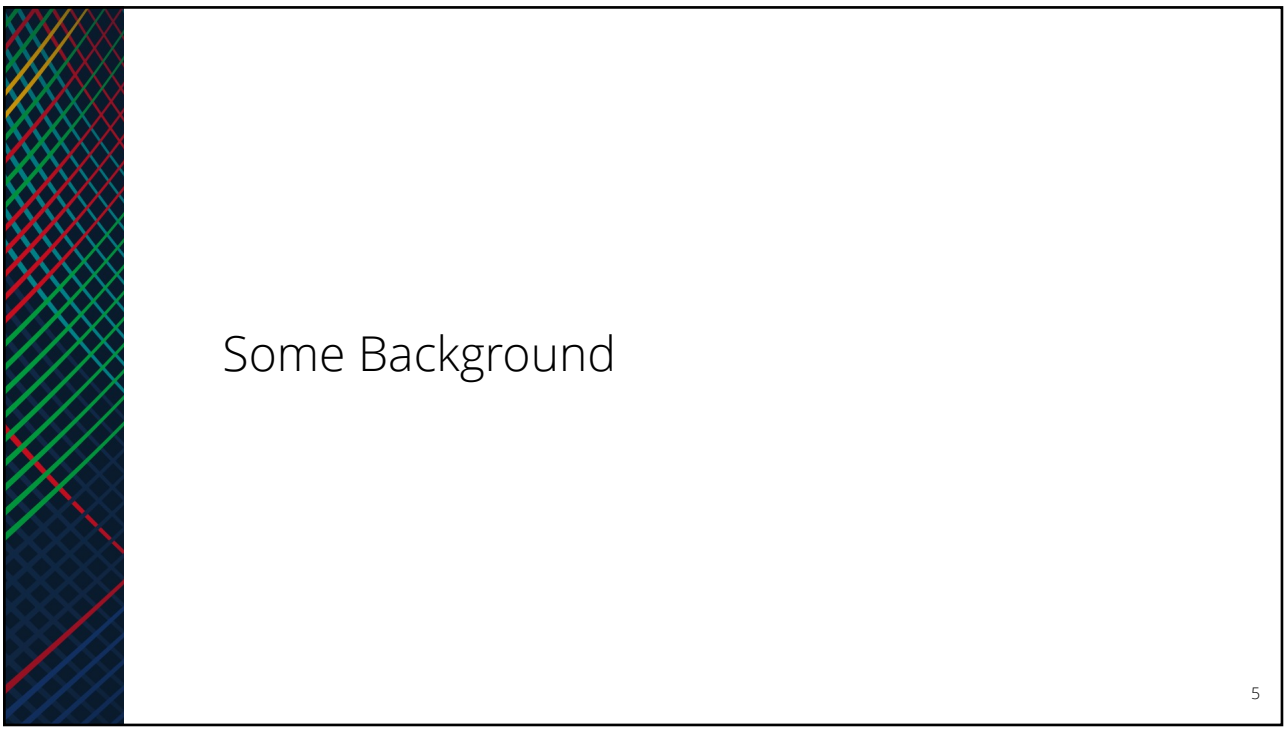
2


Motivation for Digital to Analog
3

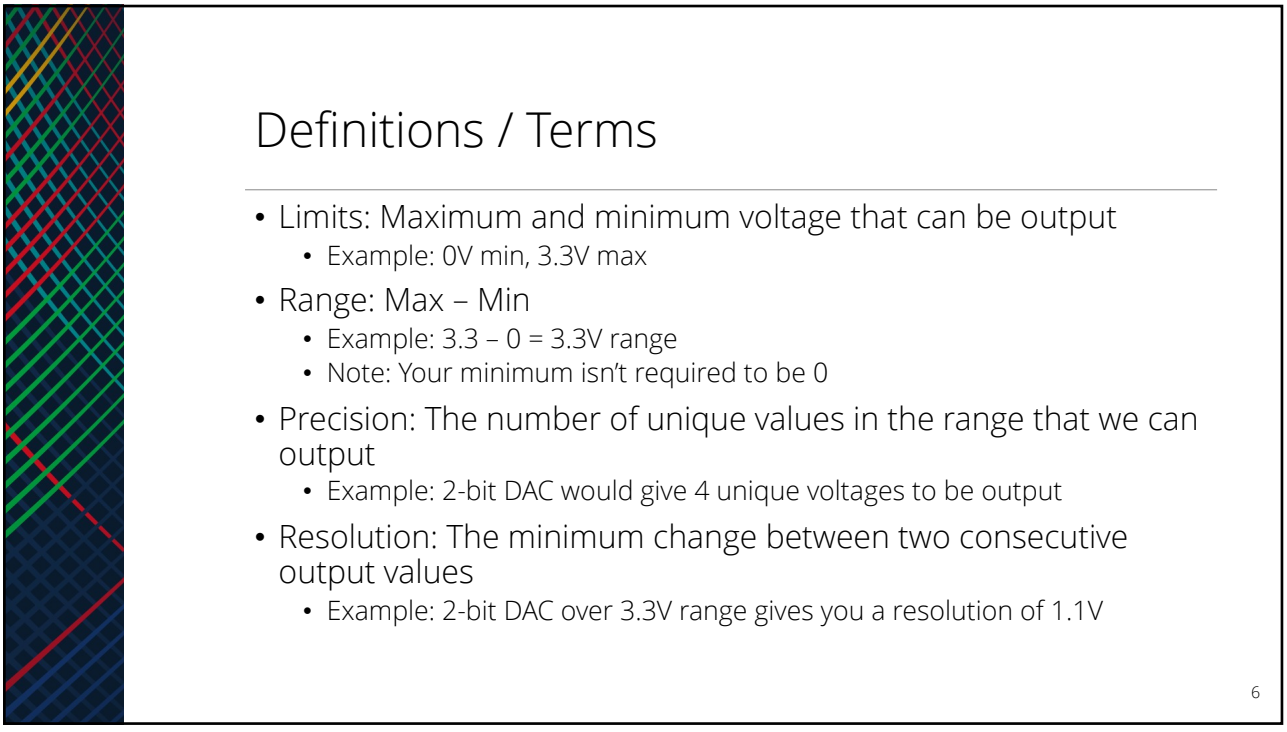
3


Why Digital to Analog?
• The world around us is analog while computers are digital
• We need to interact with the world
• Create sound waves
• Drive DC motors
• Etc.
• We need a way to produce analog signals with specific voltage levels
• But in the digital world, we can only produce Hi and Lo
4

4

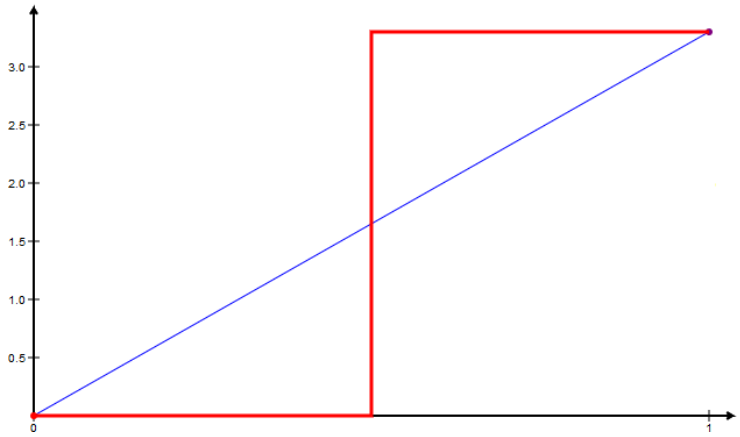

Some Background
5

5


Definitions / Terms
• Limits: Maximum and minimum voltage that can be output
• Example: 0V min, 3.3V max
• Range: Max – Min
• Example: 3.3 – 0 = 3.3V range
• Note: Your minimum isn't required to be 0
• Precision: The number of unique values in the range that we can output
• Example: 2-bit DAC would give 4 unique voltages to be output
• Resolution: The minimum change between two consecutive output values
• Example: 2-bit DAC over 3.3V range gives you a resolution of 1.1V
6

6

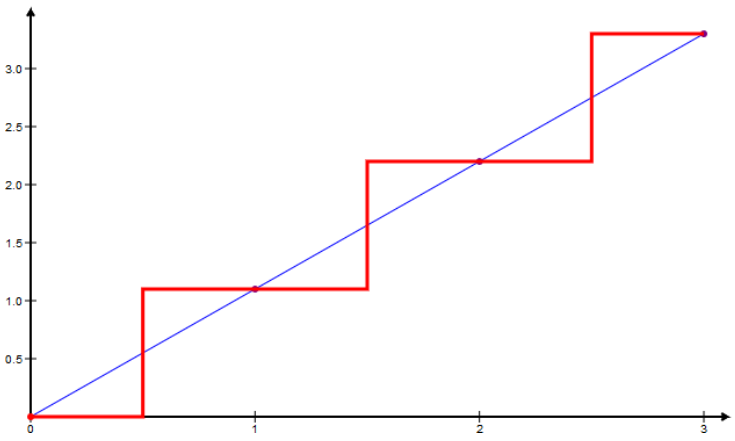
1-Bit DAC Precision



7

7

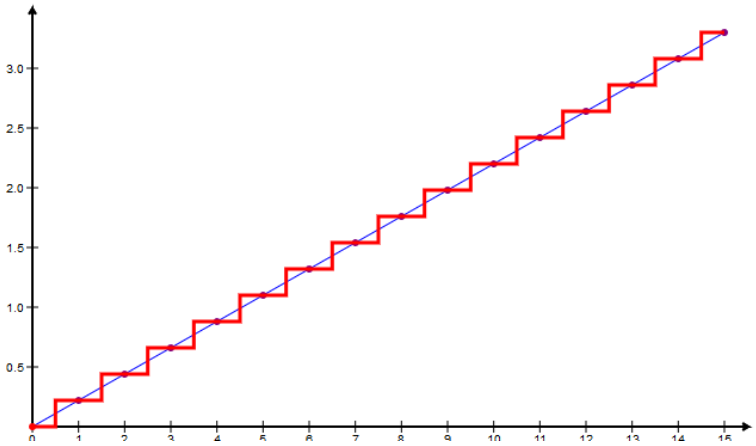
2-Bit DAC Precision



8

8

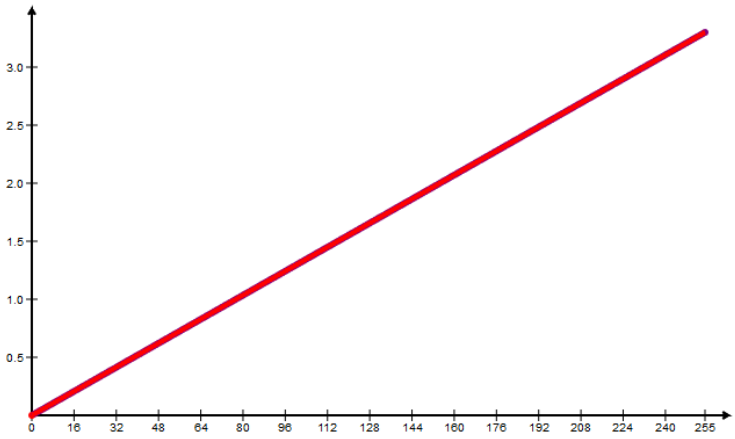
4-Bit DAC Precision



9

9

8-Bit DAC Precision



10

10

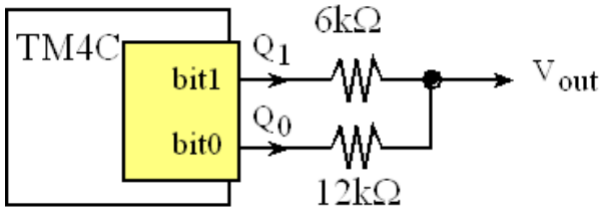
Simple DAC Design Example

Simple, 2-Bit DAC

- Let's design a 2-bit DAC such that

Q0	Q1	Output
0	0	0 V
0	1	1.1 V
1	0	2.2 V
1	1	3.3 V

2-Bit DAC Design

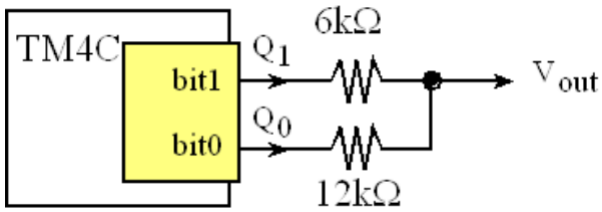


- What happens when Q0 = 0 and Q1 = 0?

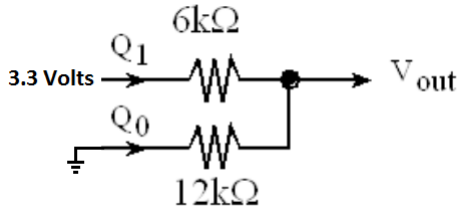
13

13

2-Bit DAC Design



- What happens when Q0 = 0 and Q1 = 1?



14

14

6-Bit DAC Design

- You can create a 6-bit DAC using 6 IO ports such that:
  - RQ0 = R
  - RQ1 = R/2
  - RQ2 = R/4
  - RQ3 = R/8
  - RQ4 = R/16
  - RQ5 = R/32
- This is called a Binary Weighted DAC

15

15

Binary Weighted DAC Limitations

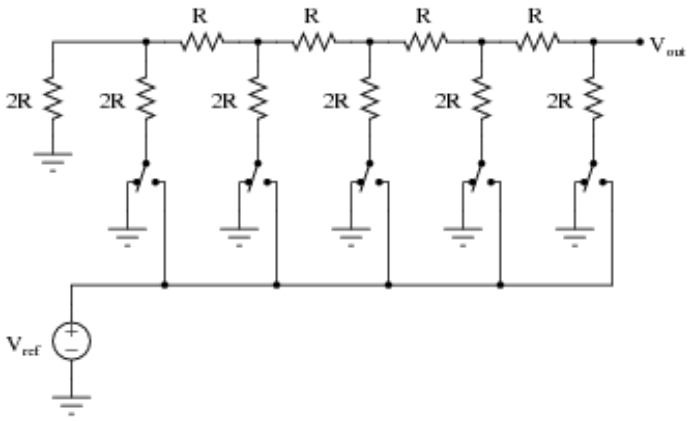
- Trouble scaling to high precision (lots of bits)
- Dependent on precise resistor values
- Example: 12-bit Binary Weighted DAC
  - Smallest resistor: 100 Ohm
  - Largest Resistor: 204,800 Ohm
  - If resistors have a 1% tolerance in manufacturing, then largest could be off by 2000 Ohm, which will mess things up

16

16

Alternative: R-2R DAC

- A design that has less range in the resistor values



17

17

Bad News

- None of these things are built into our microcontroller
- Interface an external device instead

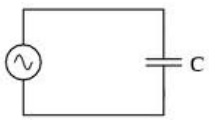
18

18

Approaches Based on PWM

Using a Capacitor

- Reminder:



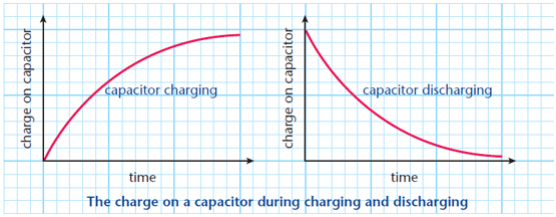


Image Source: http://www.revisionworld.com/

PWM With an RC Circuit

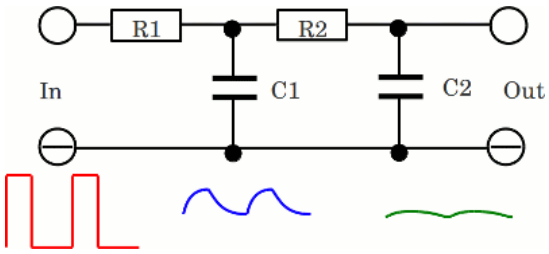


Image Source: http://www.homofaciens.de/technics-base-circuits-servos_en_navion.htm

21

21

PWM With an RC Circuit (Part 2)

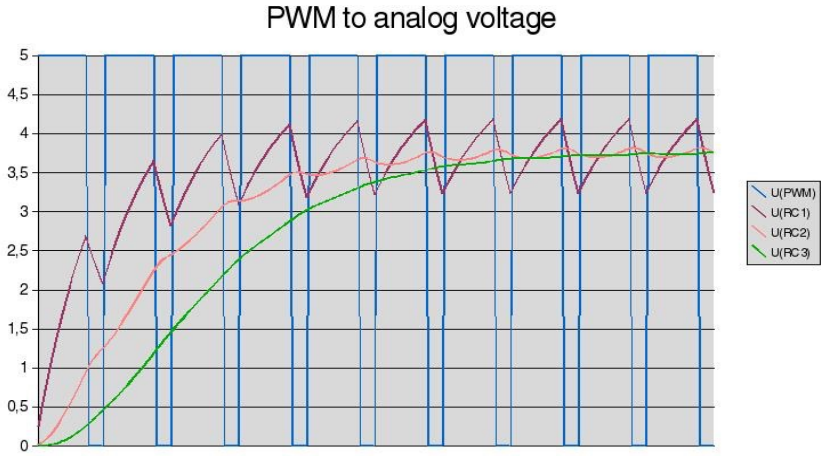


Image Source: http://www.avr-asm-tutorial.net/avr_en/AVR_ADC500.html

22

22

PWM RC Circuit Limitations

- It takes multiple stages, and and as such time, to reach a steady output voltage
- Potentially only good for ~5 outputs per second
- Precision is based on how configurable your PWM duty cycle is

23

23

Powering a DC Motor

What a *marvel*ous idea…

24

24

DC Motors

- Electric motors powered by DC voltage/current
- Apply a voltage, motor turns (Not like a servo)
- Speed is controller by the voltage More voltage -> Turns faster
- How can we control the speed from a digital system?

25

25

Controlling DC Motor Speed with PWM

- Apply a PWM signal to the motor
  - When power is applied, the motor rotates
  - When power is removed, motor runs on momentum
- Interface digital output wave with motor such that:
  - Frequency is high enough that the rotation never stops
  - The duty cycle will determine on average how much power is applied to the motor

26

26

Generating a PWM Signal

- The General Purpose Timer Modules (GPTM) can do this
  - See book section 6.3.1
- There are also dedicated PWM modules
  - See book section 6.3.2

27

27

TM4C123 PWM Modules

- TM4C123 has 2 PWM modules
- Each module has 4 blocks
- Each block produces two outputs
  - Block0 produces PWM0 and PWM1
  - Block1 produces PWM2 and PWM3
  - And so on…

28

28

PWM Pins

IO	Ain	0	1	2	3	4	5	6	7	8	9	14
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			I_2C1SCL		M1PWM2					
PA7		Port			I_2C1SDA		M1PWM3					
PB0		Port	U1Rx						T2CCP0			
PB1		Port	U1Tx						T2CCP1			
PB2		Port			I_2C0SCL				T3CCP0			
PB3		Port			I_2C0SDA				T3CCP1			
PB4	Ain10	Port		SSI2Clk		M0PWM2			T1CCP0	CAN0Rx		
PB5	Ain11	Port		SSI2Fss		M0PWM3			T1CCP1	CAN0Tx		
PB6		Port		SSI2Rx		M0PWM0			T0CCP0			
PB7		Port		SSI2Tx		M0PWM1			T0CCP1			
PC4	C1-	Port	U4Rx	U1Rx		M0PWM6		IDX1	WT0CCP0	U1RTS		
PC5	C1+	Port	U4Tx	U1Tx		M0PWM7		PhA1	WT0CCP1	U1CTS		
PC6	C0+	Port	U3Rx					PhB1	WT1CCP0	USB0epen		
PC7	C0-	Port	U3Tx						WT1CCP1	USB0pflt		
PD0	Ain7	Port	SSI3Clk	SSI1Clk	I_2C3SCL	M0PWM6	M1PWM0		WT2CCP0			
PD1	Ain6	Port	SSI3Fss	SSI1Fss	I_2C3SDA	M0PWM7	M1PWM1		WT2CCP1			
PD2	Ain5	Port	SSI3Rx	SSI1Rx		M0Fault0			WT3CCP0	USB0epen		
PD3	Ain4	Port	SSI3Tx	SSI1Tx				IDX0	WT3CCP1	USB0pflt		
PD6		Port	U2Rx			M0Fault0		PhA0	WT5CCP0			
PD7		Port	U2Tx					PhB0	WT5CCP1	NMI		
PE0	Ain3	Port	U7Rx									
PE1	Ain2	Port	U7Tx									
PE2	Ain1	Port										
PE3	Ain0	Port										
PE4	Ain9	Port	U5Rx		I_2C2SCL	M0PWM4	M1PWM2			CAN0Rx		
PE5	Ain8	Port	U5Tx		I_2C2SDA	M0PWM5	M1PWM3			CAN0Tx		
PF0		Port	U1RTS	SSI1Rx	CAN0Rx		M1PWM4	PhA0	T0CCP0	NMI	C0o	
PF1		Port	U1CTS	SSI1Tx			M1PWM5	PhB0	T0CCP1		C1o	TRD1
PF2		Port		SSI1Clk		M0Fault0	M1PWM6		T1CCP0			TRD0
PF3		Port		SSI1Fss	CAN0Tx		M1PWM7		T1CCP1			TRCLK
PF4		Port					M1Fault0	IDX0	T2CCP0	USB0epen		

29

29

PWM Configuration Registers

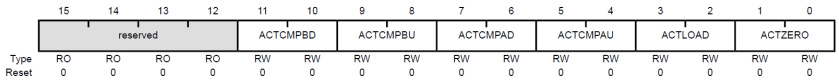
- PWM0_0_CTL_R
  - Enable a PWM output
- PWM0_0_LOAD_R
  - Period for PWM
- PWM0_0_CMPA_R
  - Duty cycle
- PWM0_ENABLE_R
  - Enable PWM0-7

30

30

PWM Configuration Registers

- PWM0_0_CFG_R



  - ACTZERO
    - What should happen when timer reaches 0
    - 0 = Do nothing
    - 1 = Invert
    - 2 = Drive low
    - 3 = Drive high
  - ACTLOAD
    - What should happen when timer is reloaded
  - ACTMPAD
    - What should happen when timer reaches Comparator A