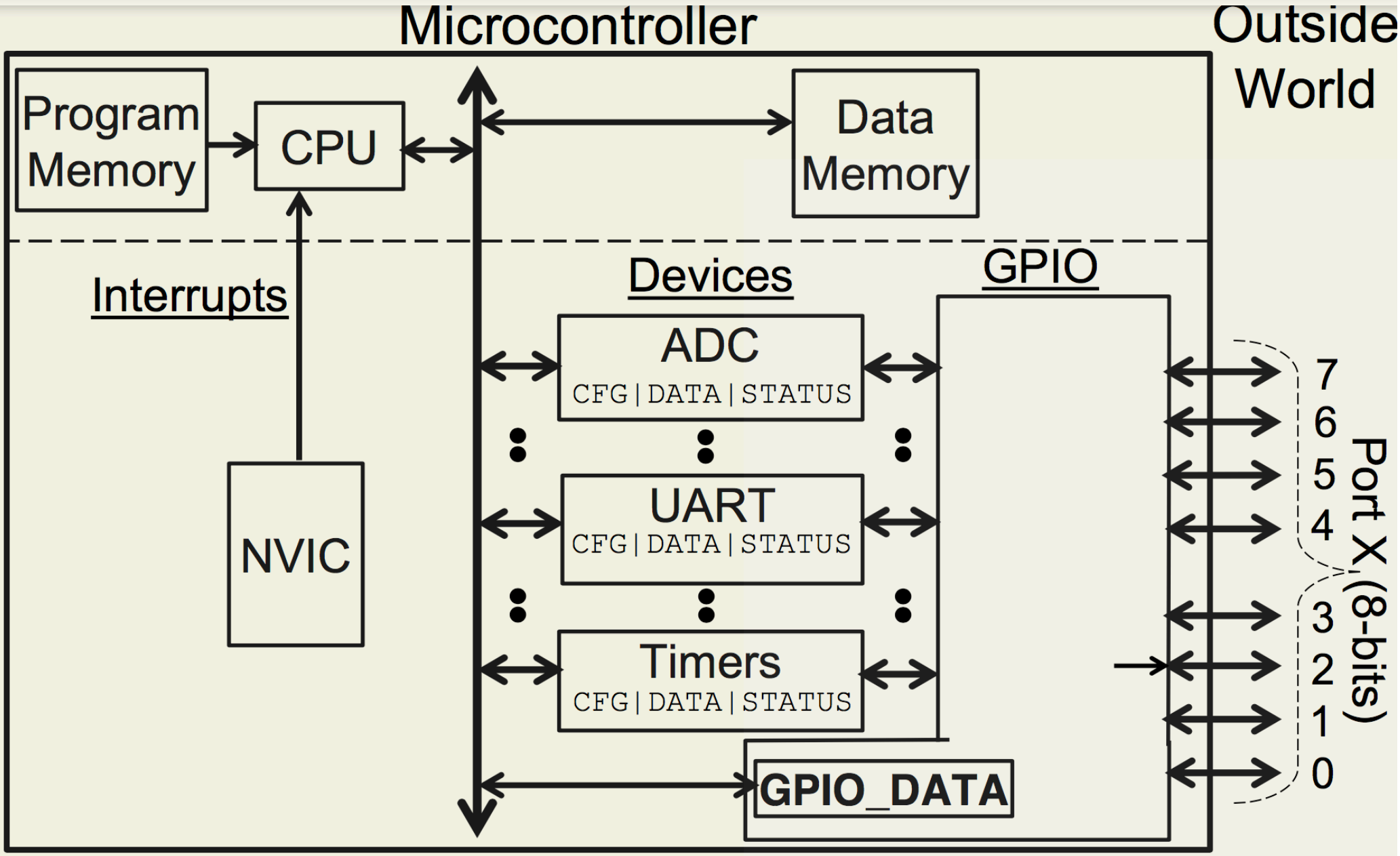
CprE 288 – Introduction to Embedded Systems (Analog to Digital Conversion Concepts, Introduction to Lab 7, Programming Interface)

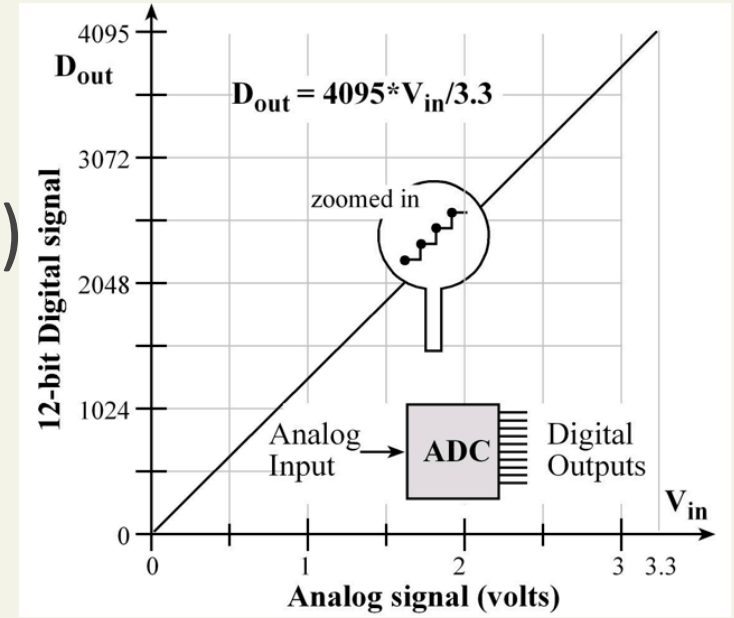
Instructor:
Dr. Diane Rover

TM4C Microcontroller



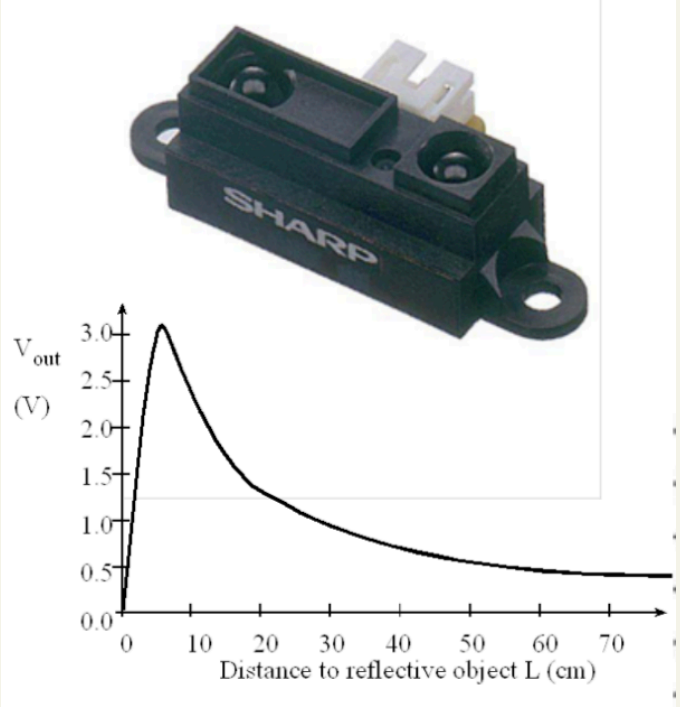
ADC Concepts

- analog-to-digital conversion (ADC)
  - analog
  - digital
- analog input channel
- sampling (data acquisition/capture)
- quantization
- ADC conversion terminology
  - analog voltage range or span
  - digital range and number of digital steps
  - digital step size, or resolution (analog voltage)
    - bit weight of least significant bit of digital result
  - n-bit resolution for an n-bit ADC

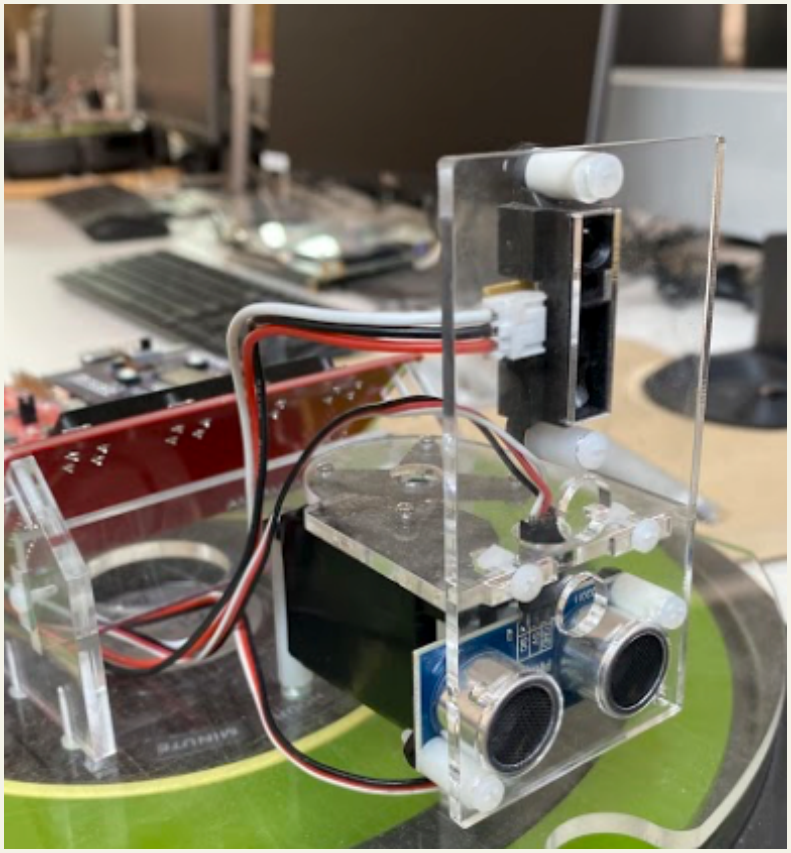


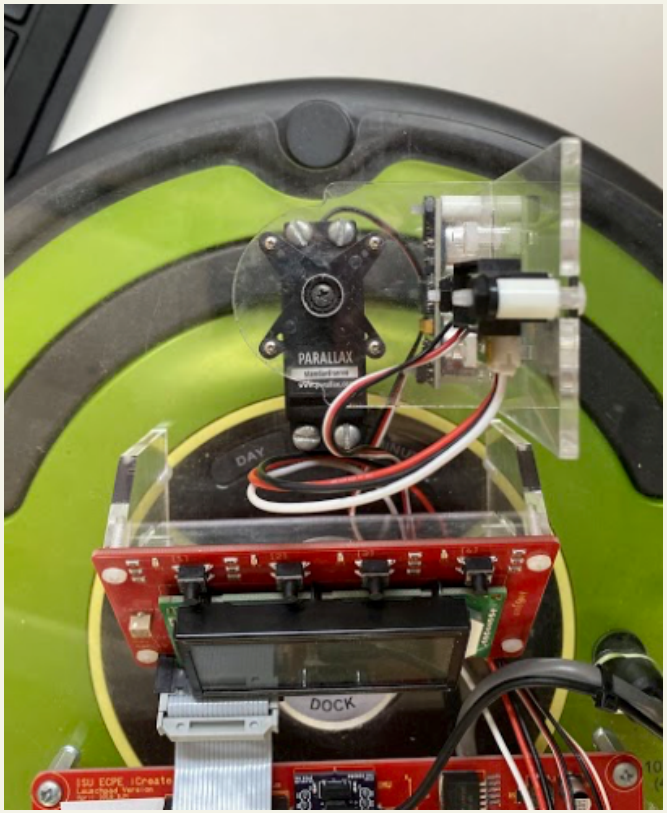
ADC Concepts (continued)

- ADC: linear operation or relationship
- sample sequencer (hardware unit in ADC module)
- sample averaging (e.g., hardware averager)
- successive approximation (ADC circuit design)
- infrared sensor
- nonlinear operation (of sensors)
- DAC: digital-to-analog conversion or converter



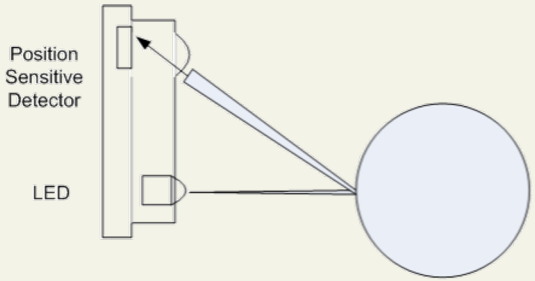
CyBot Sensor Mount and Wiring Photos

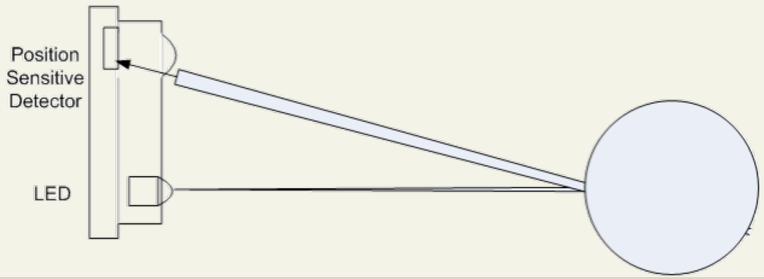


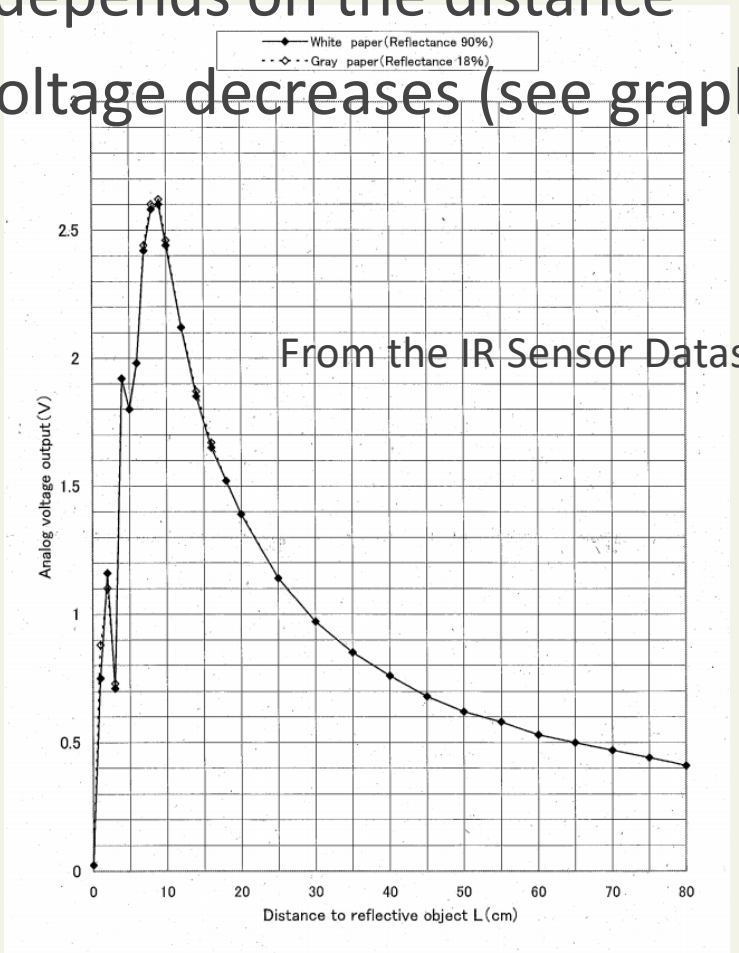


Measuring Distance with the IR Sensor

- The IR sensor emits an IR beam, and measures a voltage based on the distance of an object
  - The voltage from the IR sensor depends on the distance
  - As the distance increases, the voltage decreases (see graph)







From the IR Sensor Datasheet

How To Measure Distance with the IR Sensor

Getting a distance from the IR sensor involves the following process:

1. The IR sensor generates a voltage signal on a wire connected to a GPIO pin configured as an analog input channel
2. The ADC converts this voltage into a digital value between 0 and 4095* and stores it in an ADC register
3. Your program reads the digital result from the register and calculates a distance… but how?!?

* 12-bit ADC: 4095 = 0b1111 1111 1111 = 0xFFF

How To Measure Distance with the IR Sensor

Two methods to calculate estimated distance

1. Measure 50 points, create a look-up table
   - Create a table that has the value of the digital result when an object is X centimeters away
   - Use this table to look up the distance when a similar digital result is returned

2. Measure 5 points, use Excel to find an equation for a curve that fits the data

ADC Overview

- ADC concepts and general knowledge
  - Terminology
  - Quantization and sampling
  - Conversion formulas
  - ADC design: successive approximation
  - Performance and other issues
- TM4C123G ADC programming interface
  - GPIO initialize
  - ADC initialize
  - Reading ADC (polling vs interrupts)
- API functions you will create in lab
  - `ADC_init()`
  - `ADC_read()`

Terminology

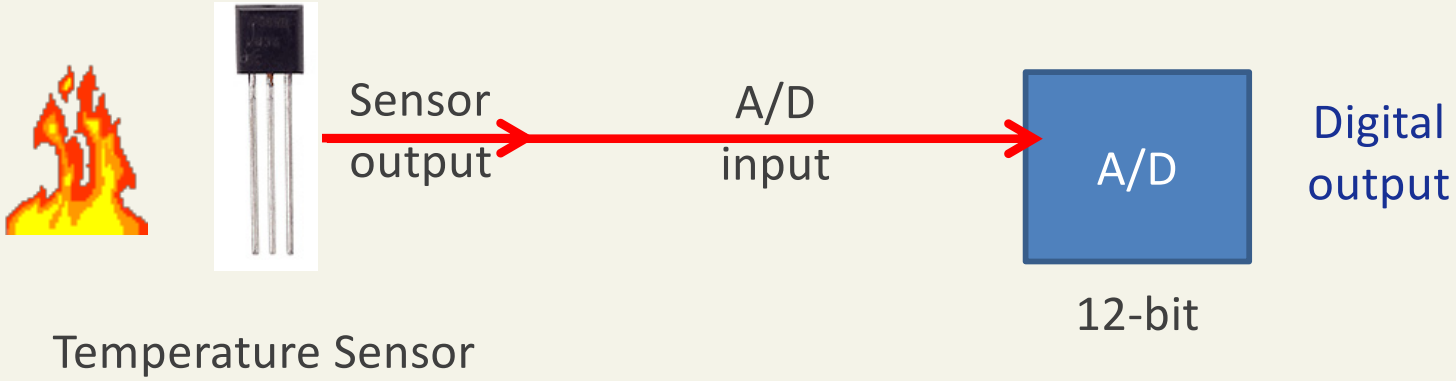
- **analog**: continuously valued signal, such as temperature, speed, or voltage with infinite possible values in between
  - E.g., between 1 m/s and 2 m/s, you can have 1.583… m/s
- **digital**: discretely valued signal, such as integers encoded in binary
  - E.g., a 2-bit integer can have only four values: 00, 01, 10, 11

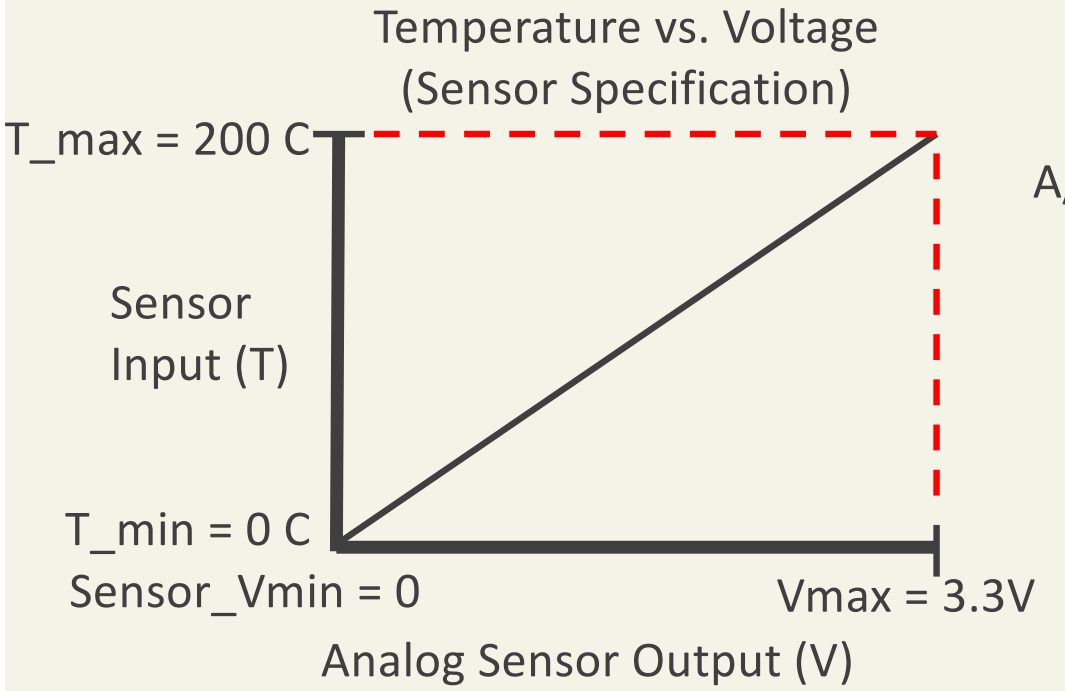
- **analog-to-digital converter** (ADC, A/D, or A2D): converts an analog input signal to a n-bit digital output signal
  - The TM4C123G has a 12-bit ADC
- **digital-to-analog converter** (DAC, D/A, D2A): converts a n-bit digital input signal to an analog output signal

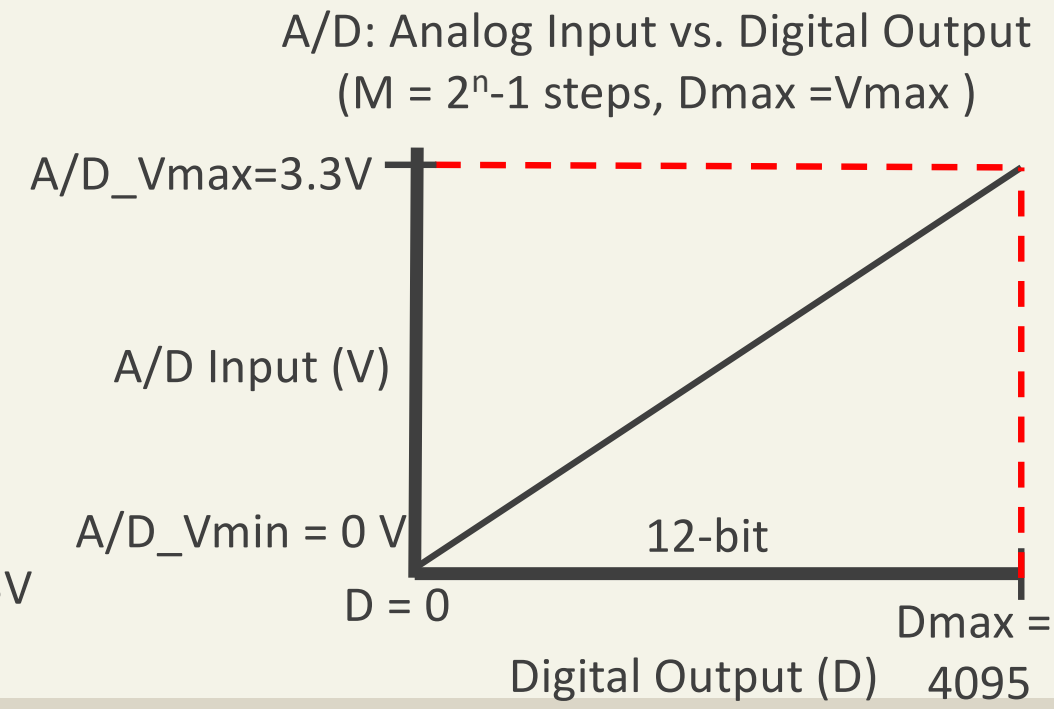
Terminology (continued)

- **span** (or **range**): difference between maximum and minimum analog values (max – min)
- **n**: number of bits used for a digital value (sometimes referred to as n-bit resolution)
- **M**: number of digital steps or values, either 2^n-1 or 2^n
- **step size** (or **resolution**): smallest analog change resulting from a change of one in a digital value; also the bit weight of the Least Significant Bit (LSB)
  - step size (or resolution) = span / M
- **bit weight**: analog value corresponding to a bit position in a digital value
- **sensitivity**: amount sensor output changes for a change in sensor input

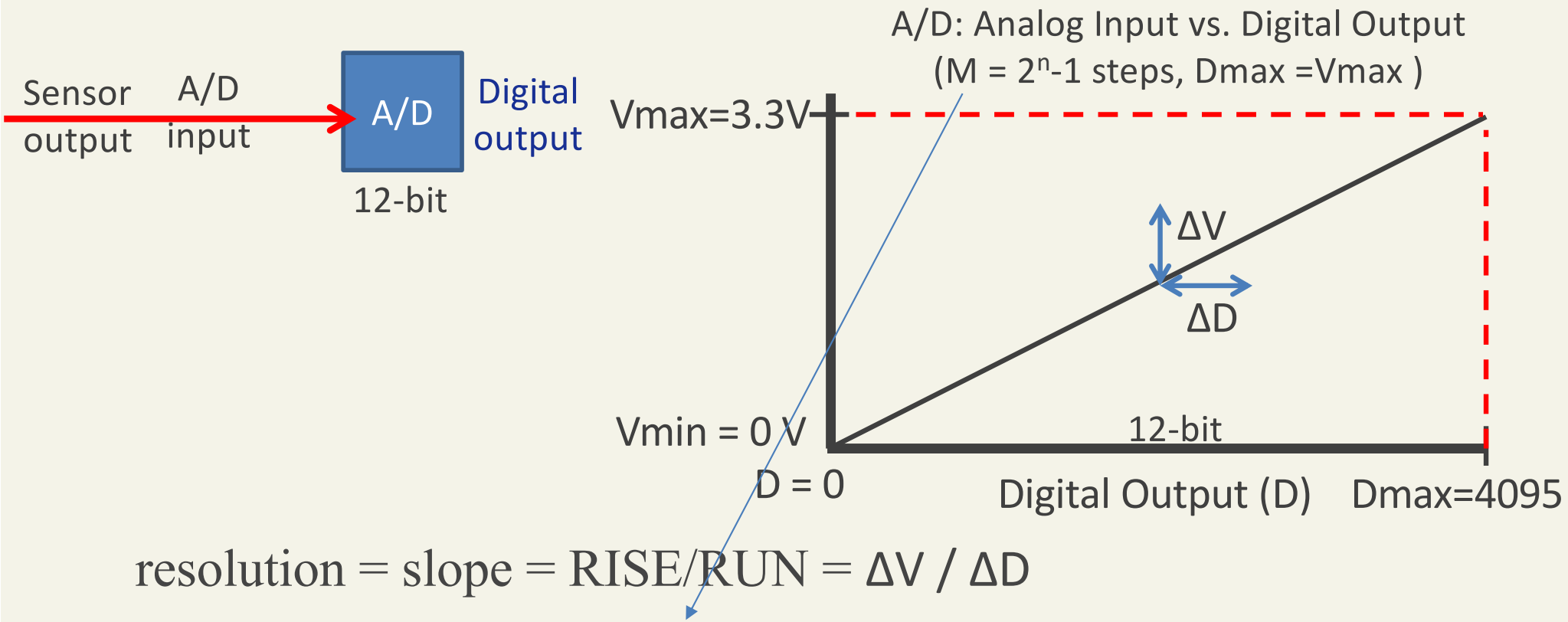
Analog-to-Digital Converter: Example







Analog-to-Digital Converter: Example



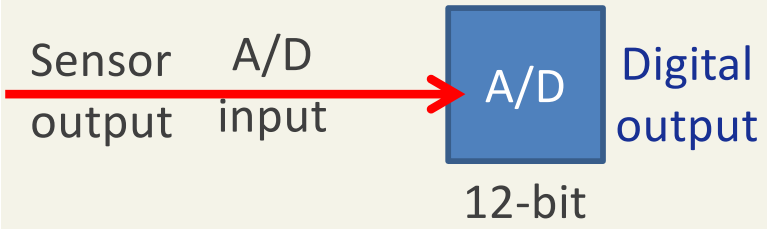
$$\text{resolution} = \text{slope} = \text{RISE/RUN} = \Delta V / \Delta D$$

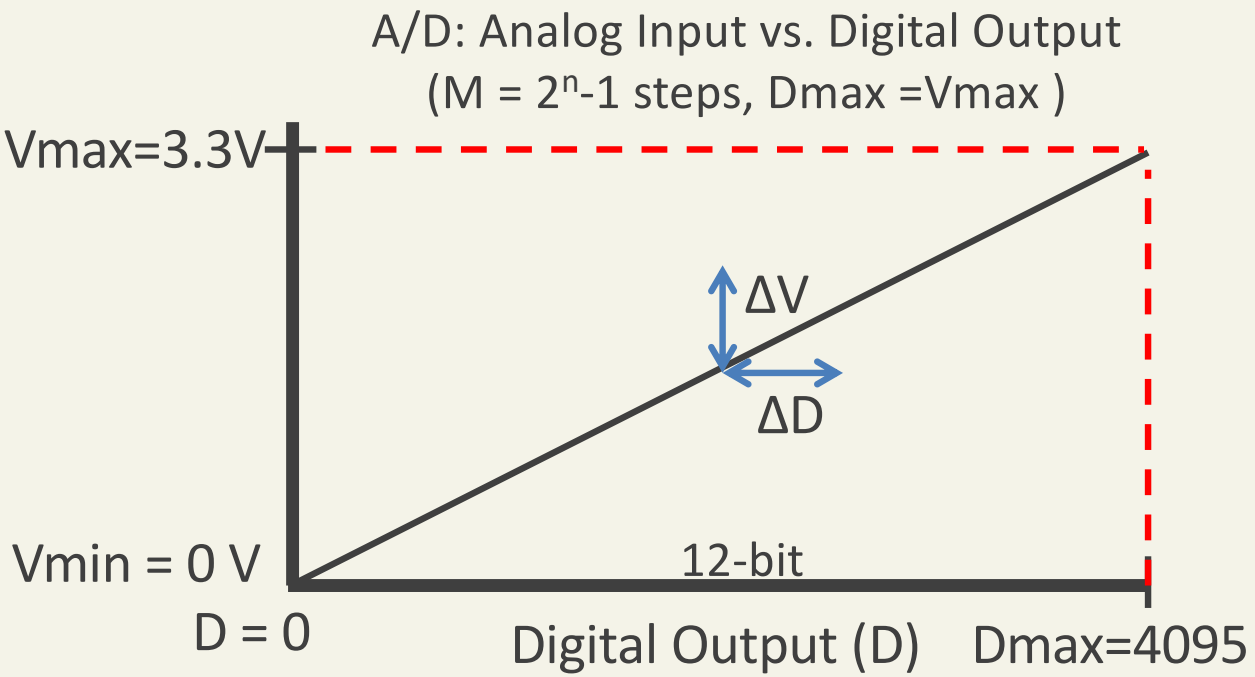
$$= (3.3 - 0 / 4095 - 0)$$

$$= 0.0008059 \text{ V/bit}$$

$$\text{LSB bit weight} = 0.0008059 \text{ V/bit}$$

Analog-to-Digital Converter: Example

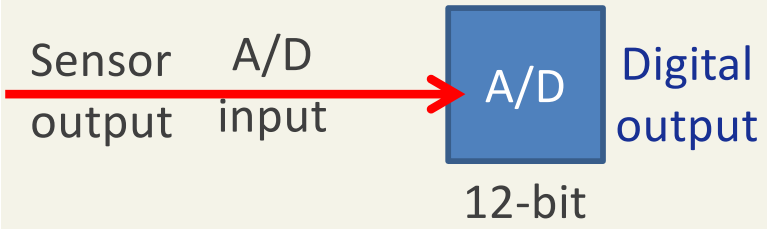


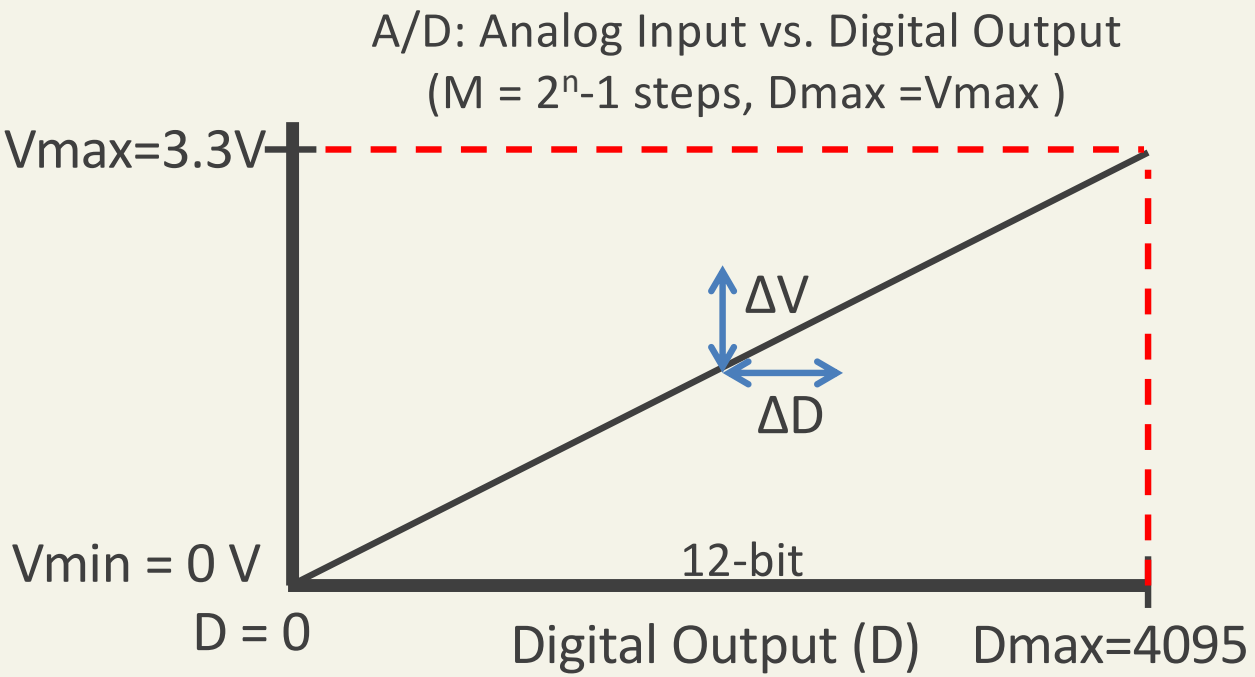


Question: If the input is 1.65 V, what is the digital output of the A/D?

Hint: A/D converters are typically linear

Analog-to-Digital Converter: Example





Y = rX + b; in this case the y-intercept b = 0, so Y = rX

We are given the voltage (Y) is 1.65V and computed r = 0.0008059, so

1.65 = .0008059X; X = 1.65/.0008059= 2047.4

Truncate to 2047 and use 12 bits = 0b0111 1111 1111 = 0x7FF

For this example, a temperature of 100 C gives a digital value of 2047

ADC Bit Weight

LSB bit weight in the last example:
bit 0 = 0.0008059V, i.e., resolution

Each bit position in digital value is weighted with an analog value, such that a 1 in that bit position adds its analog value to the total analog value represented by the digital encoding.

For the previous example:

Decimal: 2047
Binary: 0111 1111 1111
Sum of bit weights from 1..11
= 1.6497 V ~= 1.65V

Digital Bit	Bit Weight (V)
11	2048*r = 1.6505
10	1024*r = 0.8252
9	512*r = 0.4126
8	256*r = 0.20638
7	128*r = 0.1031
6	64*r = 0.05158
5	32*r = 0.02579
4	16*r = 0.01289
3	8*r = 0.006447
2	4*r = 0.003224
1	2*r = 0.001612
0	r = 0.0008059

Proportional Signals

Assume Vmin = 0 V

Vmax = maximum voltage of the analog range

a = analog value

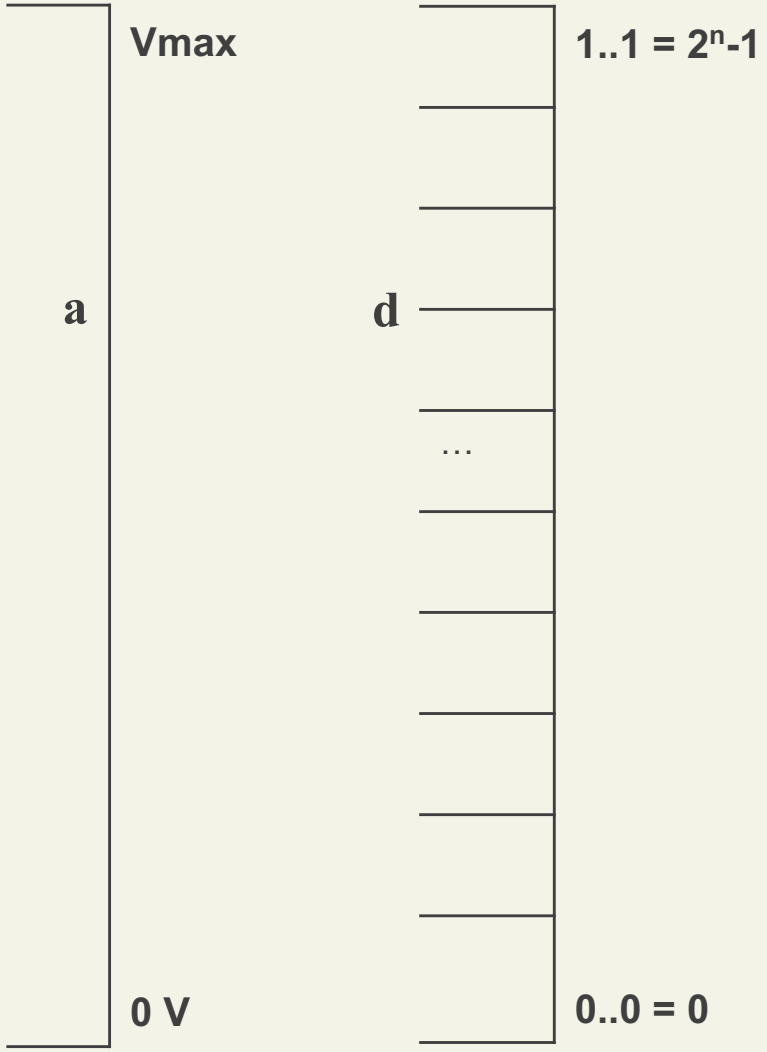
n = number of bits for digital encoding

2^n = number of digital codes

M = number of steps, either 2^n or $2^n - 1$

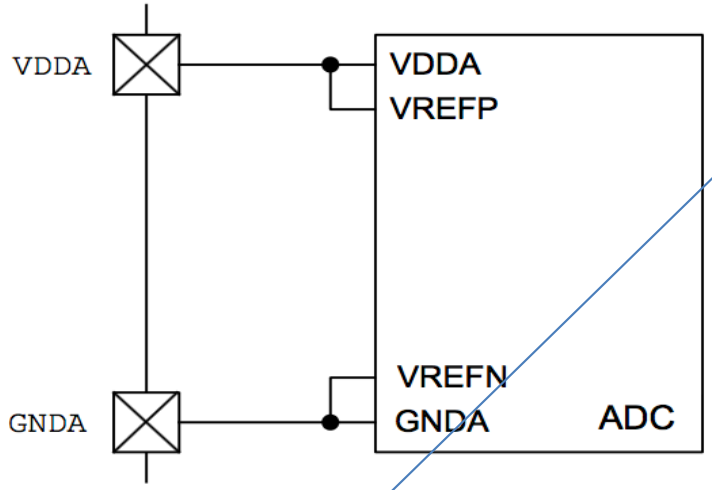
d = digital encoding

a / Vmax = d / M



ADC Module in Tiva Datasheet (Chapter 13)

Figure 13-8. ADC Voltage Reference



a / Vmax = d / M

r = Vmax / M = a / d

n = 12
2^{12} = 4096
Dmin = 0
Dmax = 4095 = 0xFFF
M = 2^n = 2^{12} = 4096 (for TM4C ADC)
Vmin = VREFN = 0 V
Vmax = VREFP = VDDA = 3.3 V

The range of this conversion value is from 0x000 to 0xFFF. In single-ended-input mode, the 0x000 value corresponds to the voltage level on VREFN; the 0xFFF value corresponds to the voltage level on VREFP. This configuration results in a resolution that can be calculated using the following equation:

```
mV per ADC code = (VREFP – VREFN) / 4096
```

While the analog input pads can handle voltages beyond this range, the analog input voltages must remain within the limits prescribed by Table 24-33 on page 1389 to produce accurate results. Figure 13-9 on page 810 shows the ADC conversion function of the analog inputs.

ADC Module in Tiva Datasheet

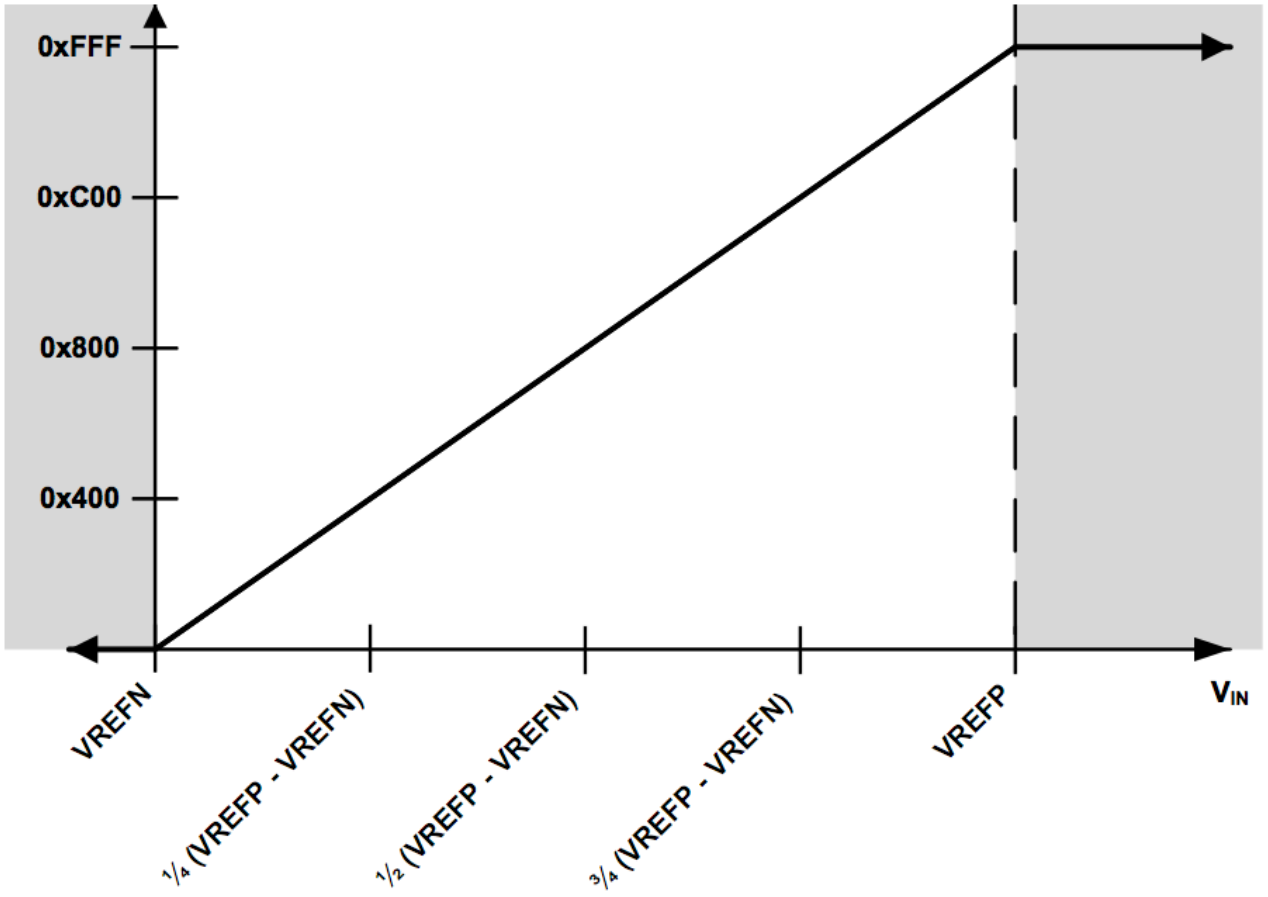
Table 24-33. ADC Electrical Characteristics[ab]

Parameter	Parameter Name	Min	Nom	Max	Unit
POWER SUPPLY REQUIREMENTS					
V_{DDA}	ADC supply voltage	2.97	3.3	3.63	V
GNDA	ADC ground voltage	-	0	-	V
VDDA / GNDA VOLTAGE REFERENCE					
C_{REF}	Voltage reference decoupling capacitance	-	1.0 // 0.01[c]	-	µF
ANALOG INPUT					
V_{ADCIN}	Single-ended, full- scale analog input voltage, internal reference[de]	0	-	V_{DDA}	V
	Differential, full-scale analog input voltage, internal reference[df]	$-V_{DDA}$	-	V_{DDA}	V
VIN_{CM}	Input common mode voltage, differential mode[g]	-	-	(VREFP + VREFN) / 2 ± 25	mV
I_L	ADC input leakage current[h]	-	-	2.0	µA
R_{ADC}	ADC equivalent input resistance[h]	-	-	2.5	kΩ
C_{ADC}	ADC equivalent input capacitance[h]	-	-	10	pF
R_S	Analog source resistance[h]	-	-	500	Ω
SAMPLING DYNAMICS					
F_{ADC}	ADC conversion clock frequency[i]	-	16	-	MHz
F_{CONV}	ADC conversion rate		1		Msps
T_S	ADC sample time	-	250	-	ns
T_C	ADC conversion time[j]		1		µs
T_{LT}	Latency from trigger to start of conversion	-	2	-	ADC clocks

ADC Module in Tiva Datasheet

Analog-to-Digital Converter (ADC)

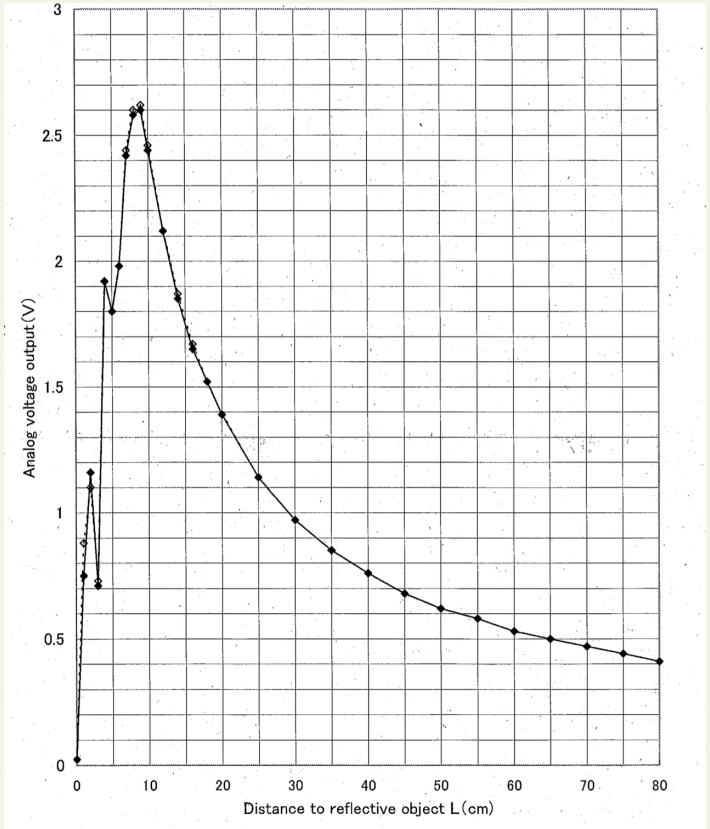
Figure 13-9. ADC Conversion Result



IR Sensor & ADC Input-Output Characteristics

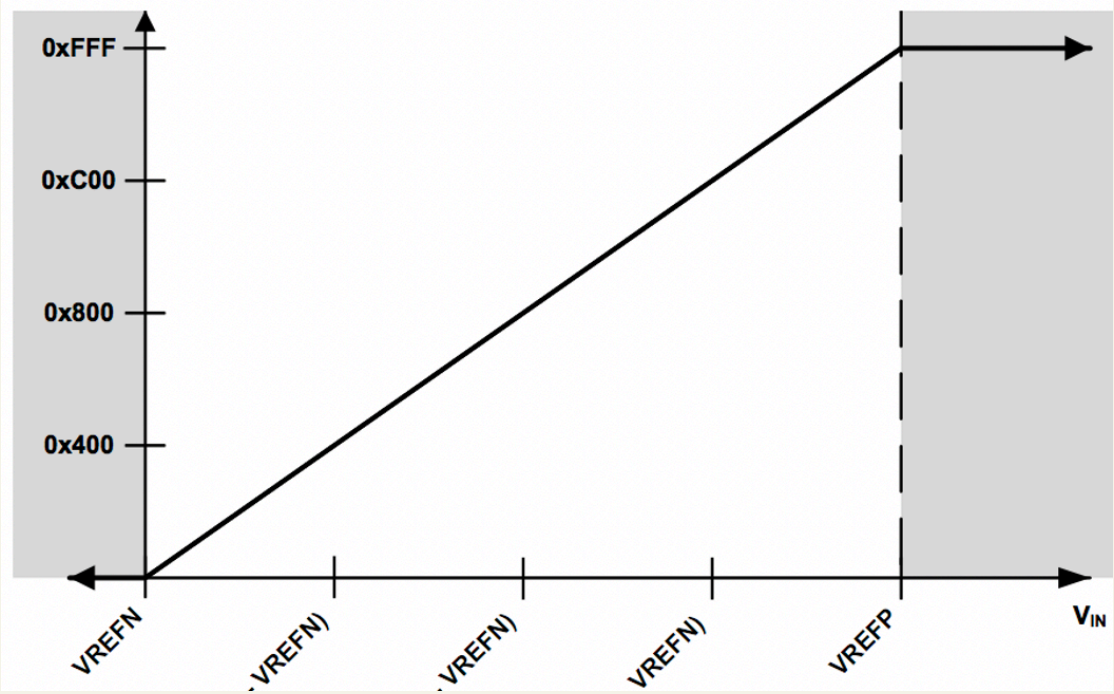
Sensor input: distance to object in cm (d_o)

→ Sensor output: analog voltage (a)



→ ADC input: analog voltage from sensor (a)

→ ADC output: digital result (d)



Resolution Illustration: $M = 2^n - 1$ vs. 2^n

Let n = 2

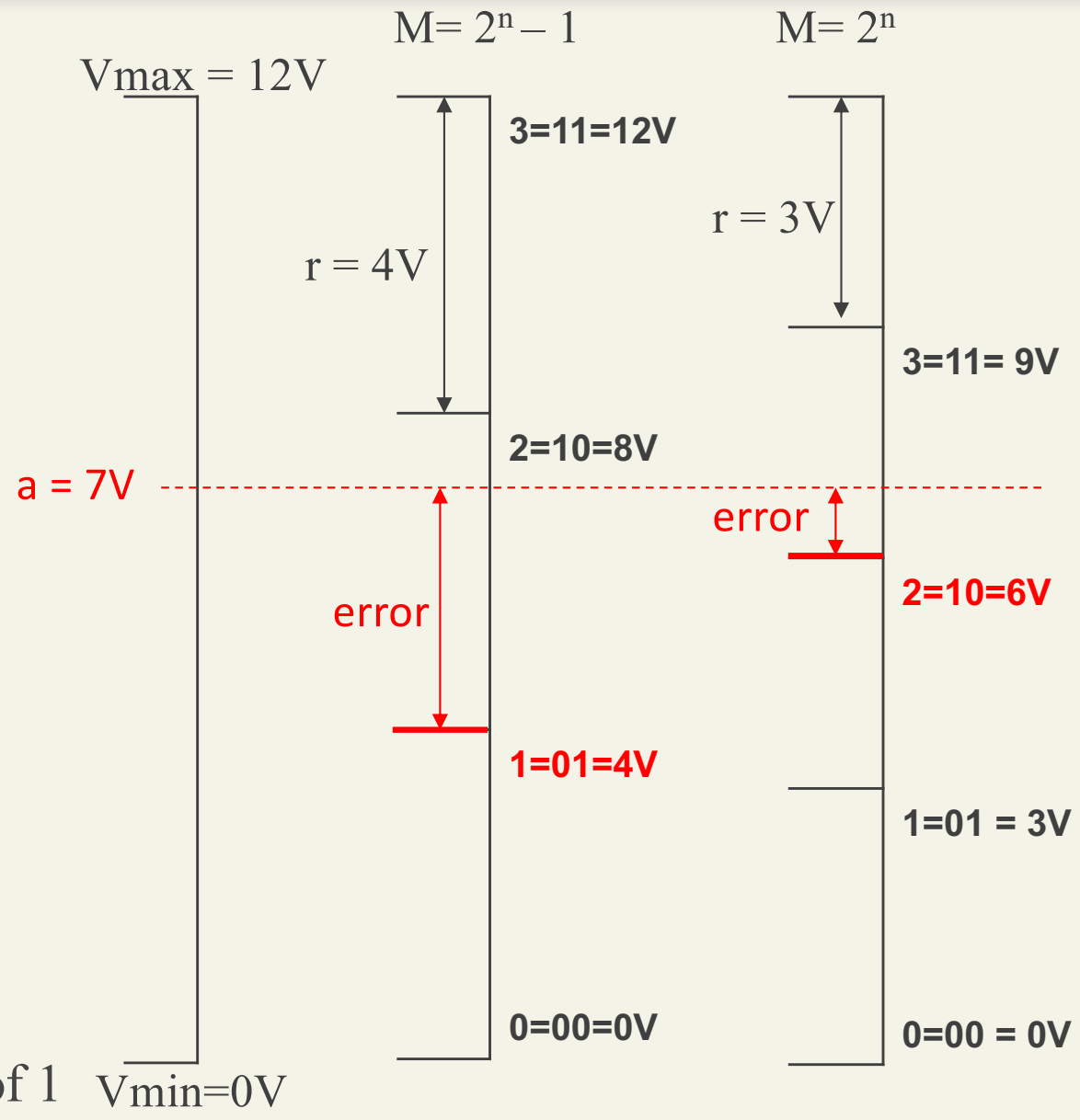
$\underline{M = 2^n - 1}$
3 steps on the digital scale

$d_0 = 0 = 0b00$
$d_{Vmax} = 3 = 0b11$

$\underline{M = 2^n}$
4 steps on the digital scale

$d_0 = 0 = 0b00$
$d_{Vmax - r} = 3 = 0b11$ (no d_{Vmax}, it would be at 0b100=4)

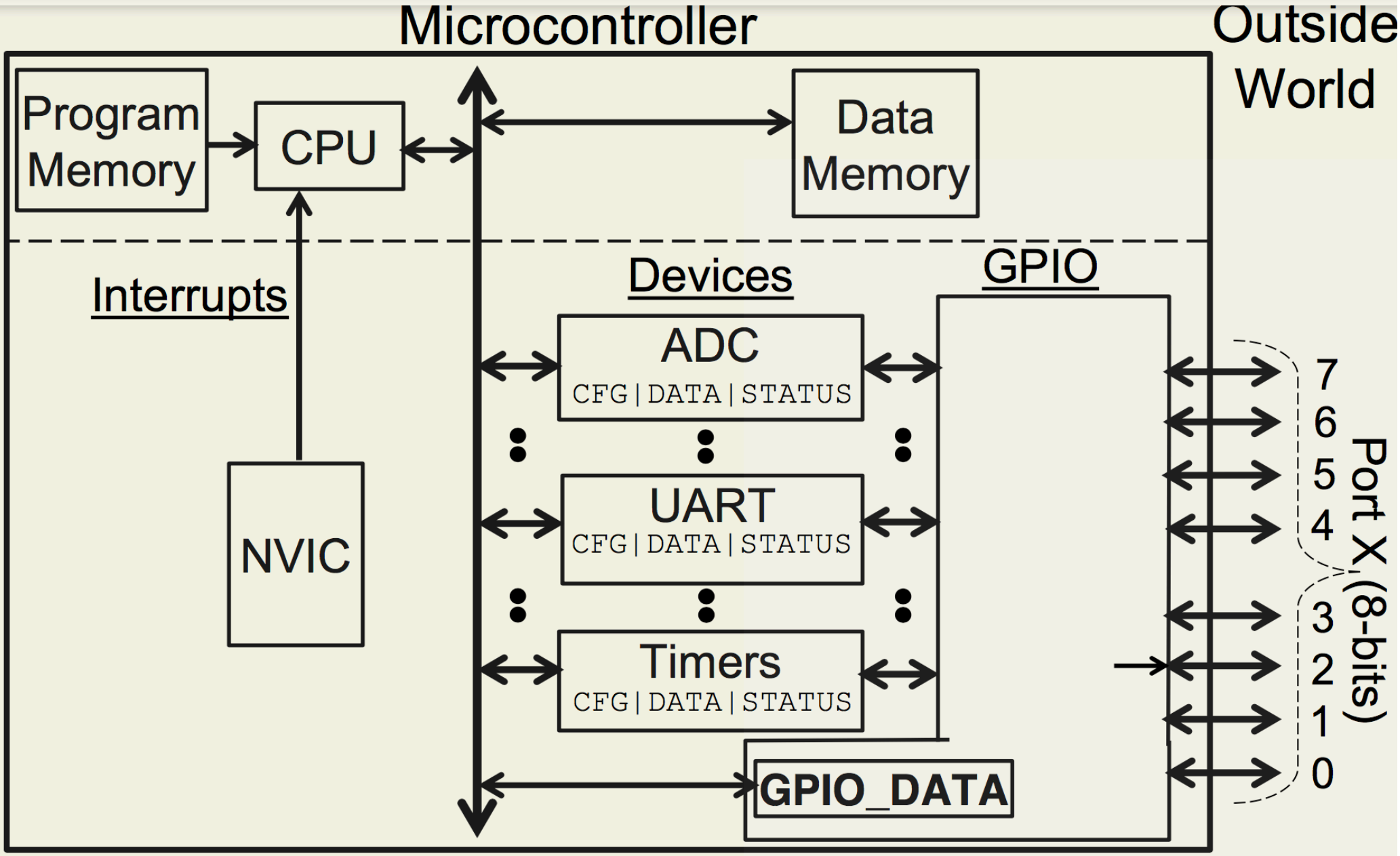
r, **resolution**: analog change resulting from a digital change of 1



ADC Overview

- ADC concepts and general knowledge
  - Terminology
  - Quantization and sampling
  - Conversion formulas
  - ADC design: successive approximation
  - Performance and other issues
- TM4C123G ADC programming interface
  - GPIO initialize
  - ADC initialize
  - Reading ADC (polling vs interrupts)
- API functions you will create in lab
  - `ADC_init()`
  - `ADC_read()`

TM4C Microcontroller



ADC Programming

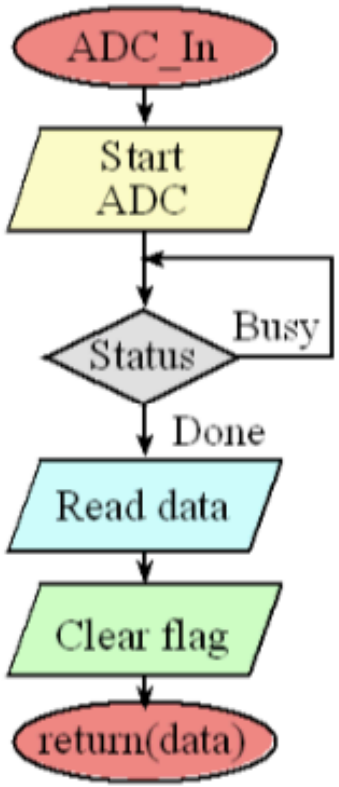


Figure 14.3. The four steps of analog to digital conversion: 1) initiate conversion, 2) wait for the ADC to finish, 3) read the digital result, and 4) clear the completion flag.

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C14_ADCdataAcquisition.htm

ADC Input Channels

Table 7.3 ADC input channels and GPIO pins distributions

ADC Pin	GPIO Pin	Pin Type	Pin Function
AIN0	**PE3**	Input	ADC Analog Input Channel 0
AIN1	**PE2**	Input	ADC Analog Input Channel 1
AIN2	**PE1**	Input	ADC Analog Input Channel 2
AIN3	**PE0**	Input	ADC Analog Input Channel 3
AIN4	**PD3**	Input	ADC Analog Input Channel 4
AIN5	**PD2**	Input	ADC Analog Input Channel 5
AIN6	**PD1**	Input	ADC Analog Input Channel 6
AIN7	**PD0**	Input	ADC Analog Input Channel 7
AIN8	**PE5**	Input	ADC Analog Input Channel 8
AIN9	**PE4**	Input	ADC Analog Input Channel 9
AIN10	**PB4**	Input	ADC Analog Input Channel 10
AIN11	**PB5**	Input	ADC Analog Input Channel 11

7.5.2 ADC Module Architecture and Functional Block Diagram

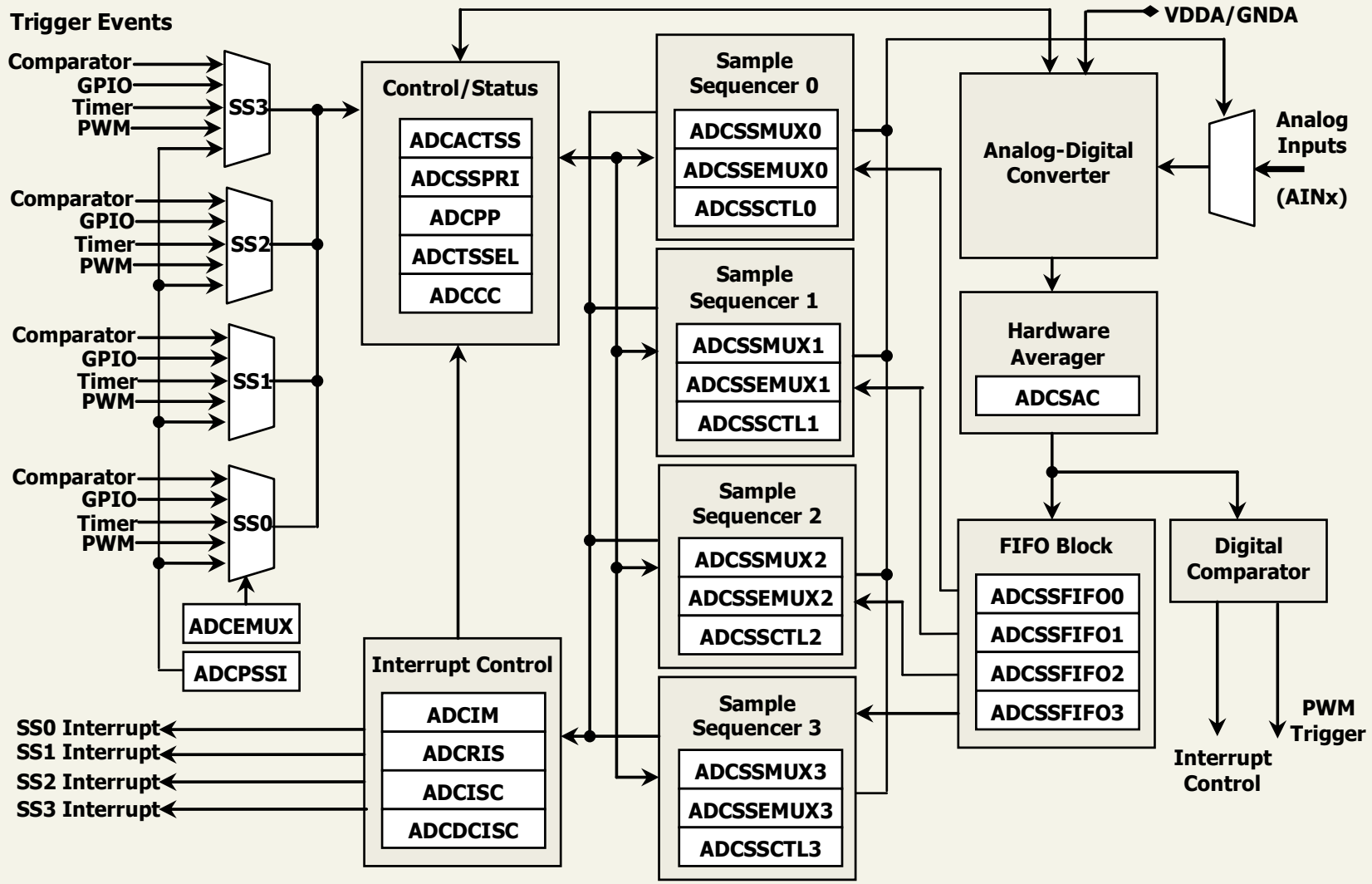


Figure 7.8 Simplified ADC module functional block diagram.

7.5.2 ADC Module Architecture and Functional Block Diagram

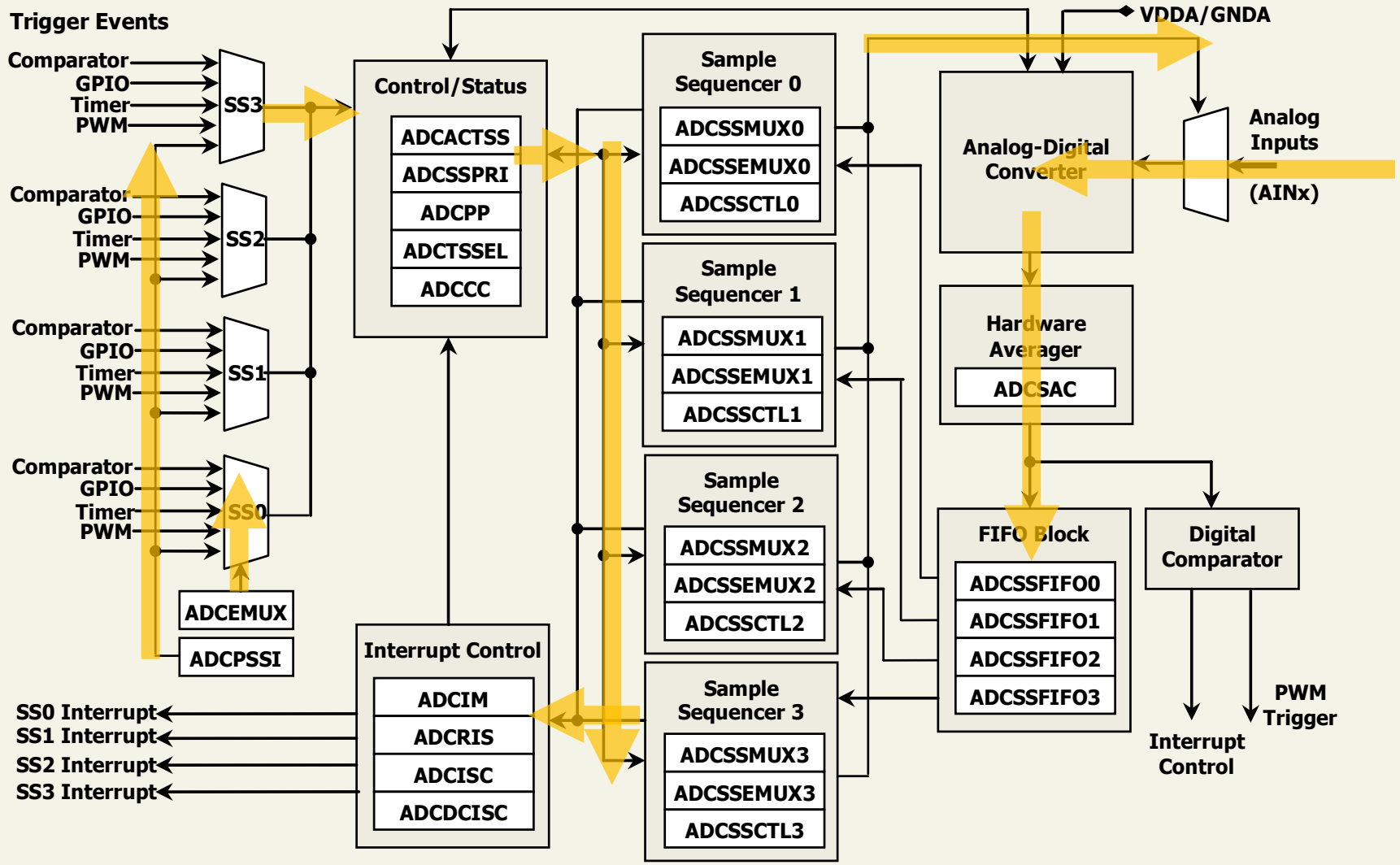
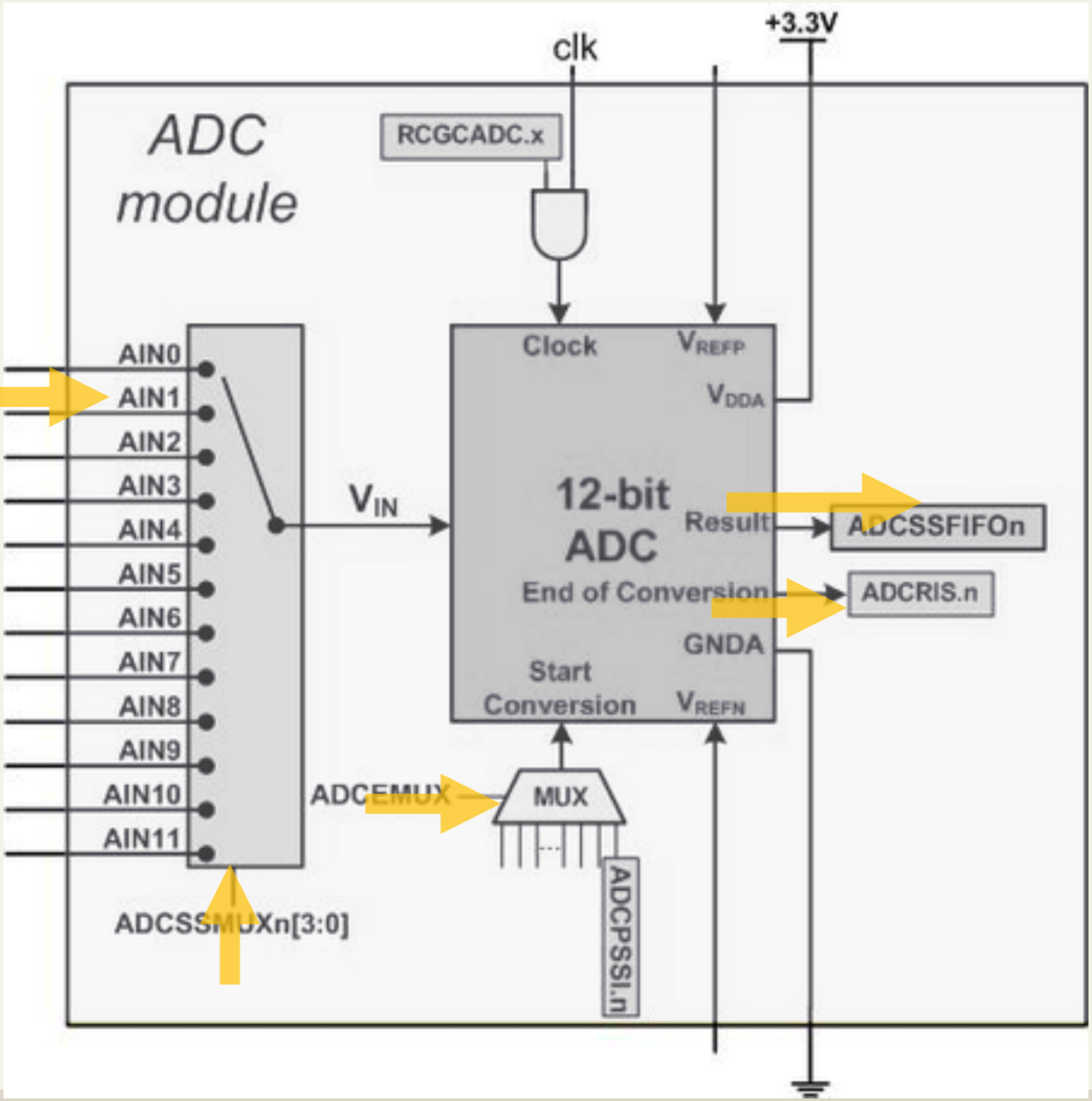


Figure 7.8 Simplified ADC module functional block diagram.

ADC Module and Sample Sequencer



Differences between Sample Sequencers

- Each ADC module contains **4** sample sequencers, **SS0** ~ **SS3**.
- These sample sequencers can be programmed to pre-define the way and the function for each sample via some registers in the related **Sample Sequencer n block**. Each sample can be obtained from different input sources in different channels.
- The sampling control and data capture is handled by the sample sequencers. **All of the sequencers are identical in implementation except for the number of samples that can be captured and the depth of the FIFO.**
- Table 7.4 shows the number of samples that each sequencer can capture and its corresponding FIFO depth.

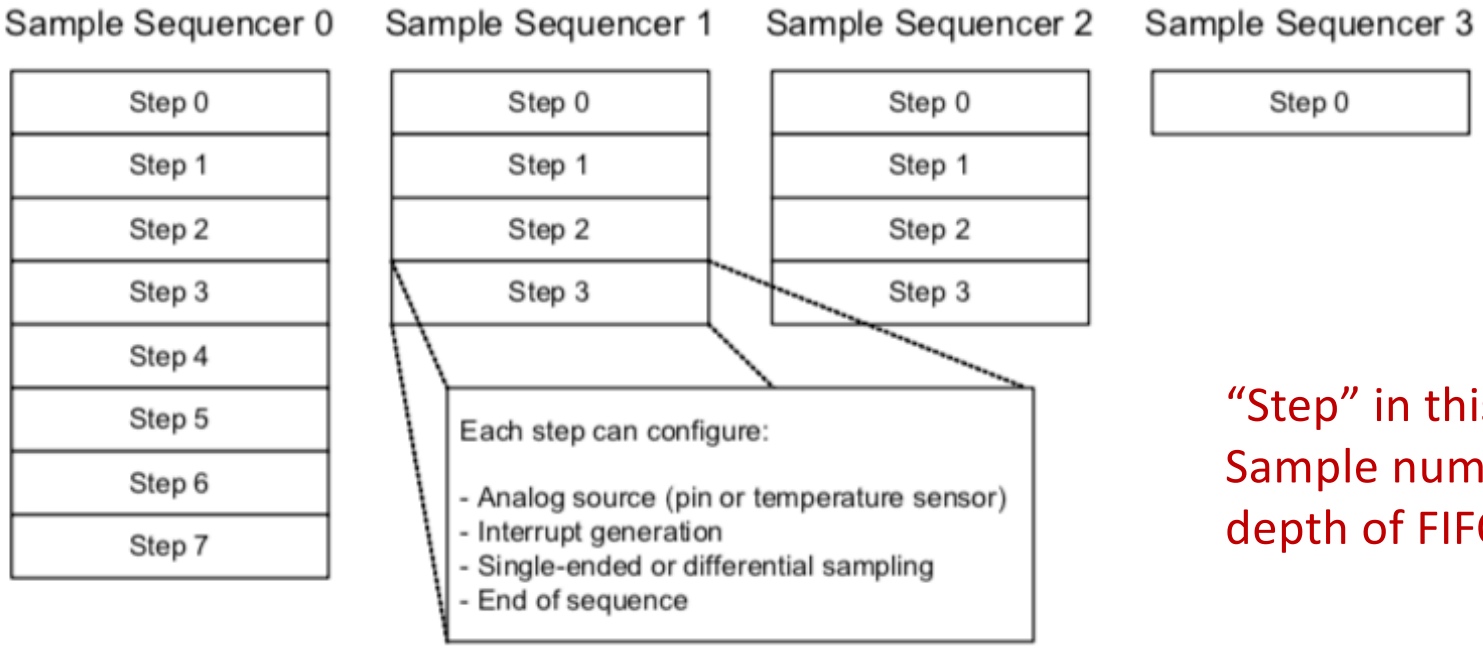
Table 7.4 Samples and FIFO depth of sequencer.

Sequencer	Number of Samples	Depth of FIFO
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

Sample Sequencer Structure

The ADC module has a total of four sample sequencers that allow sampling of 1, 4 (there are two 4-beat sequencers), or 8 analog sources with a single-trigger event (see Figure 1). Each sample sequencer has its own set of configuration registers making it completely independent from the other sequencers. All steps in a sample sequence are configurable, allowing software to select the analog input channel (including the temperature sensor), single-ended or differential mode sampling, and whether or not to generate an interrupt after the step completes. The sample sequences also have configurable priority to handle cases where multiple sequences are triggered by the same trigger source or trigger simultaneously.

Figure 1. Sample Sequencer Structure



"Step" in this figure = Sample number or depth of FIFO

ADC GPIO Registers

RCGCGPIO GPIO Run Mode Clock Gating Control

PRGPIO GPIO Peripheral Ready

GPIOAFSEL GPIO Alternate Function Select

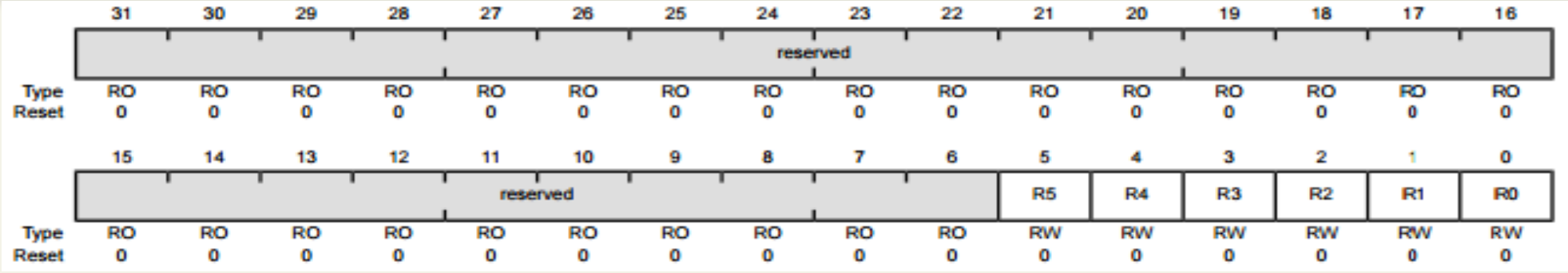
GPIODIR GPIO Direction

GPIODEN GPIO Digital Enable

GPIOAMSEL GPIO Analog Mode Select

GPIOADCCTL GPIO ADC Control

RCGCGPIO – GPIO run mode clock gating control



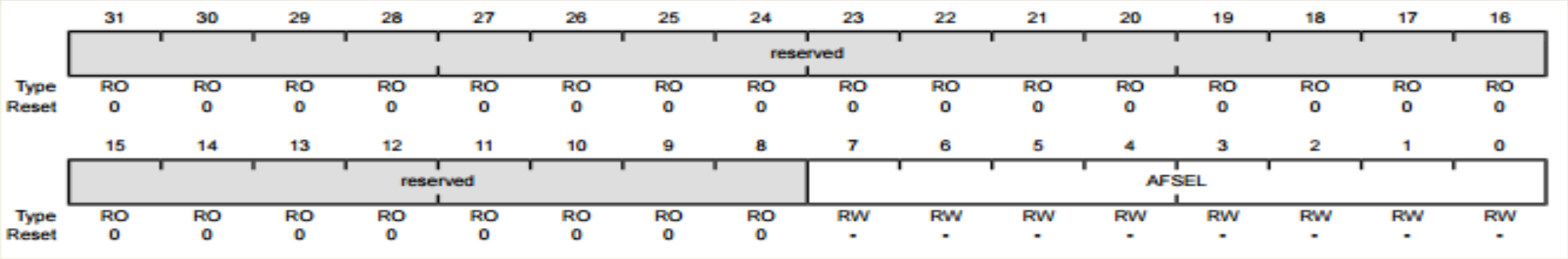
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved										R5	R4	R3	R2	R1	R0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5 R5 – 0 disable port F, 1 provide clock to port F
4 R4 – 0 disable port E, 1 provide clock to port E
3 R3 – 0 disable port D, 1 provide clock to port D
2 R2 – 0 disable port C, 1 provide clock to port C
1 R1 – 0 disable port B, 1 provide clock to port B
0 R0 – 0 disable port A, 1 provide clock to port A

In lab we will provide clock to port B

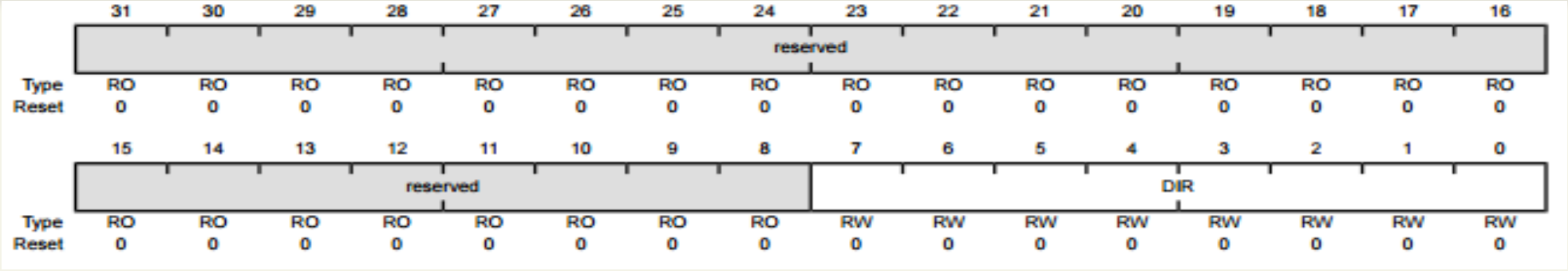
GPIOAFSEL – Alternate function select



7:0 AFSEL – 0 pin functions as normal GPIO, 1 pin works with alternate function

In lab we will set PB4 to alternate function

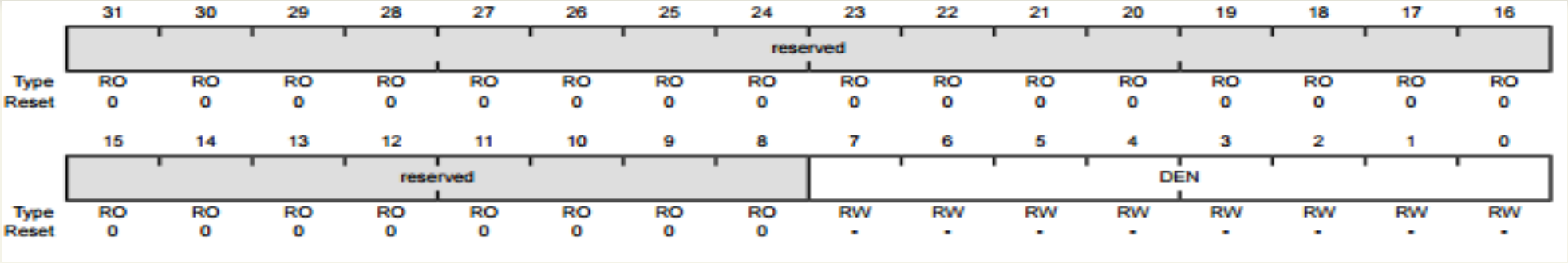
GPIODIR – GPIO direction



7:0 DIR – 0 pin is input, 1 pin is output

In lab we will set PB4 to input

GPIODEN – GPIO digital enable



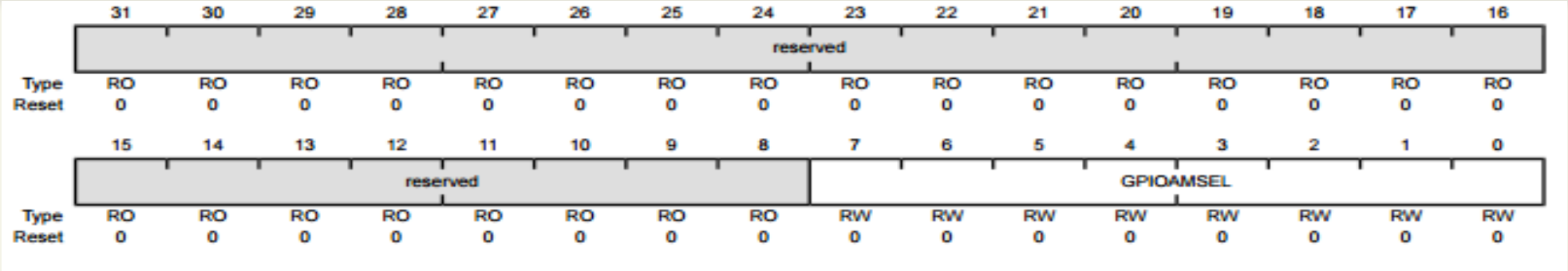
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								DEN							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

7:0 DEN – 0 digital function disabled, 1 digital function enabled

In lab we will disable PB4 digital function

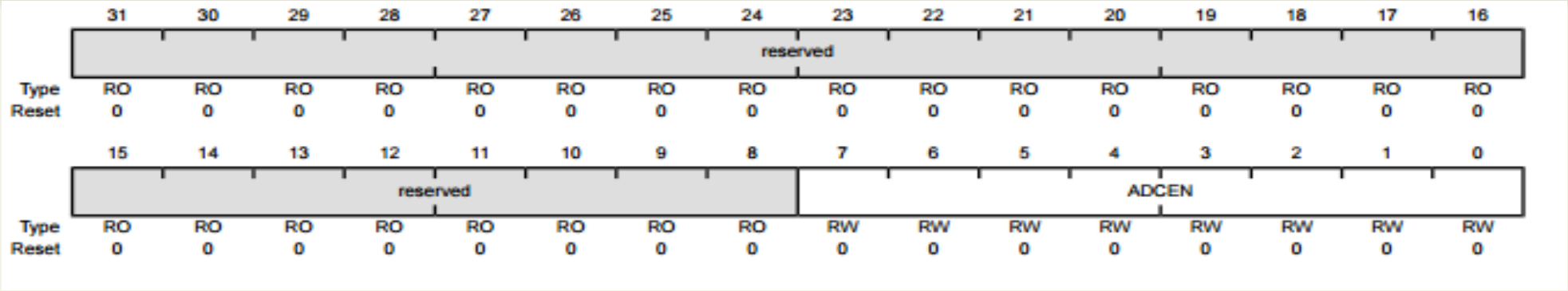
GPIOAMSEL – Analog mode select



7:0 GPIOAMSEL – 0 analog function disabled, 1 analog function enabled

In lab we will enable PB4 analog function

GPIOADCCTL – ADC control



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								ADCEN							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7:0 ADCEN – 0 = pin not used to trigger ADC, 1 = pin is used to trigger ADC

In lab we will not be using any pins to trigger ADC conversion,
so this can be written 0x00, which should be the default value

ADC Registers

RCGCADC Analog-to-Digital Converter Run Mode Clock Gating Control

PRADC ADC Peripheral Ready

ADCACTSS ADC Active Sample Sequencer

ADCEMUX ADC Event Multiplexer Select

ADCSSMUXn ADC Sample Sequence Input Multiplexer Select n (n = 0..3)

ADCSSCTLn ADC Sample Sequence Control n (n = 0..3)

ADCPSSI ADC Processor Sample Sequence Initiate

ADCRIS ADC Raw Interrupt Status

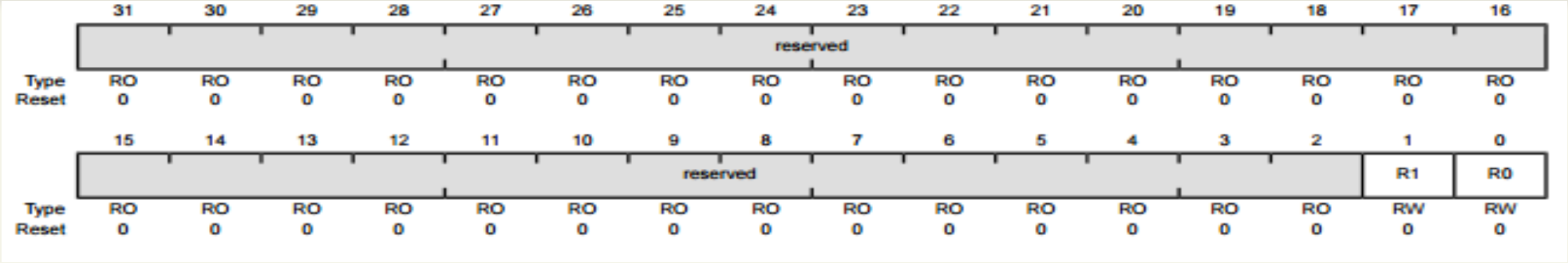
ADCSSFIFOn ADC Sample Sequence Result FIFO n (n = 0..3)

ADCISC ADC Interrupt Status and Clear

ADCSAC ADC Sample Averaging Control

Page 819 of Tiva datasheet

RCGCADC – ADC run mode clock gating control



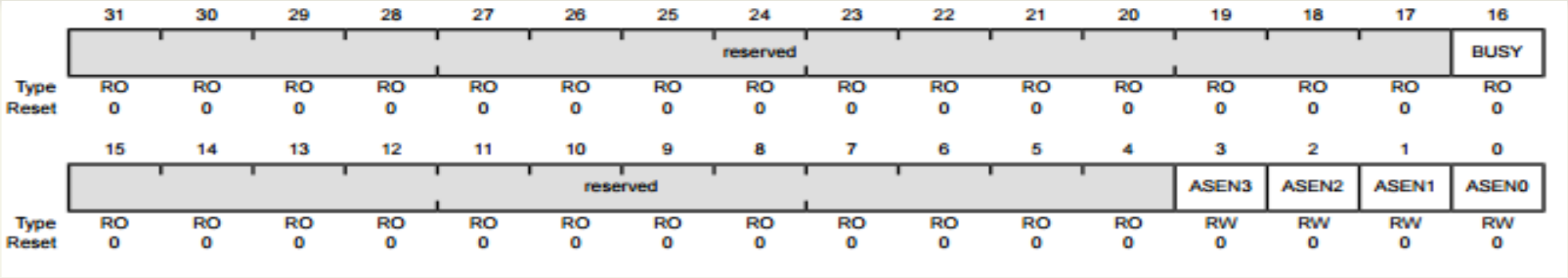
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved														R1	R0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1 R1 – 1 provide clock to ADC1, 0 disable ADC1

0 R0 – 1 provide clock to ADC0, 0 disable ADC0

ADCACTSS – Active sample sequencer

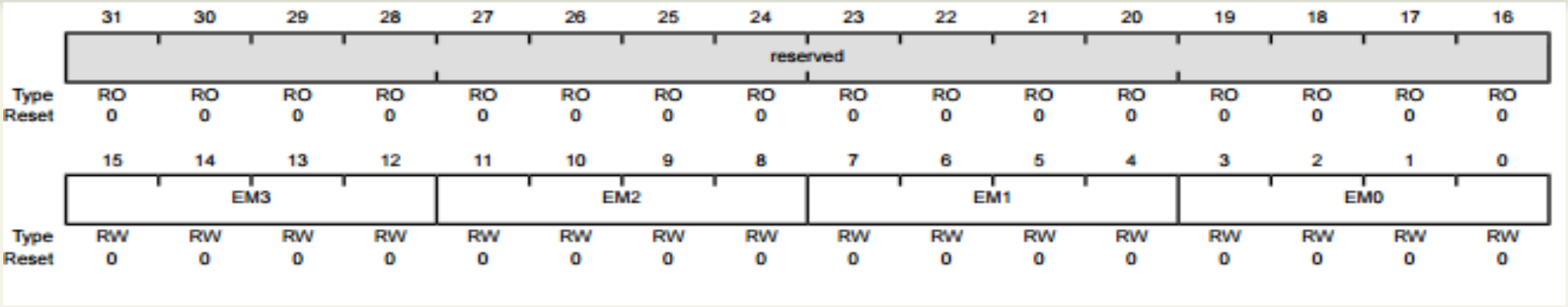


	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															BUSY
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												ASEN3	ASEN2	ASEN1	ASEN0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3 ASEN3 – 0 disable SS3, 1 enable SS3
2 ASEN2 – 0 disable SS2, 1 enable SS2
1 ASEN1 – 0 disable SS1, 1 enable SS1
0 ASEN0 – 0 disable SS0, 1 enable SS0

ADCEMUX – Event multiplexer select



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EM3				EM2				EM1				EM0			
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3:0 EM0 – SS0 trigger select
7:4 EM1 – SS1 trigger select
11:8 EM2 – SS2 trigger select
15:12 EM3 – SS3 trigger select

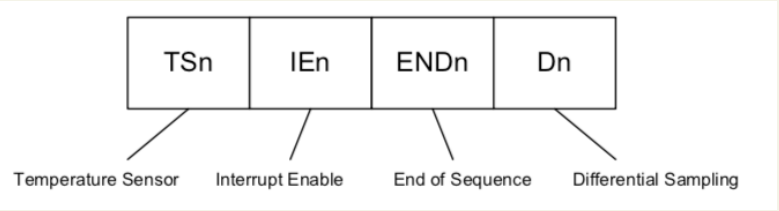
If using SS1, EM1 will be set to 0x0 using the default trigger (i.e., processor or software), which means your C program must start a conversion by writing to a memory-mapped register.

ADCSSCTLn – Sample sequence control

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS3	IE3	END3	D3	TS2	IE2	END2	D2	TS1	IE1	END1	D1	TS0	IE0	END0	D0
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3 TS0 – 1 if temp sensor 1st sample differential input select, 0 if ADCSSMUXn is read
2 IE0 – 1 if 1st sample raw interrupt enable, else 0
1 END0 – 1 if 1st sample is end of sequence, else 0
0 D0 – 1st sample differential select

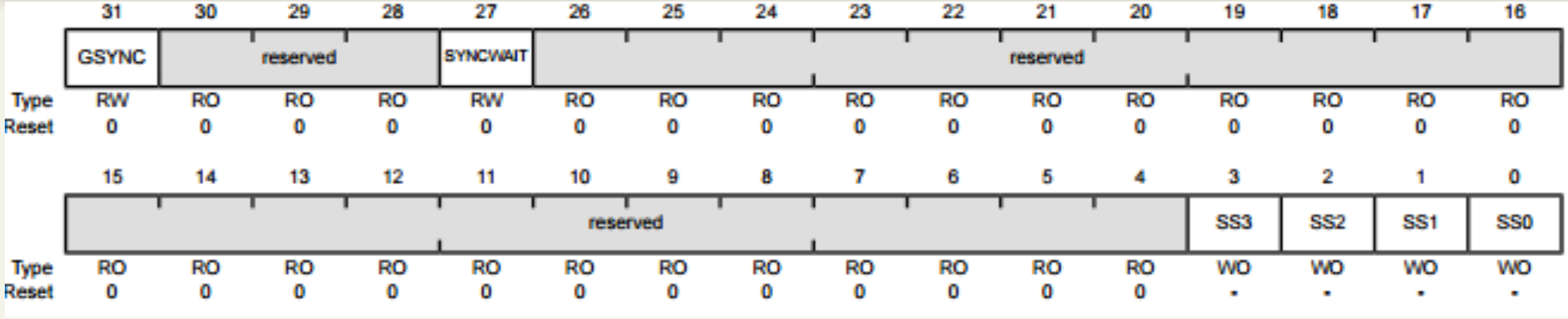


*3, *2, *1, refer to 4th, 3rd, and 2nd samples but function the same

In lab we will only need to sample once so we will end at the first sample.
We will also check the raw interrupt status.

***Note this specific register is for SS1 though others will be nearly identical; see datasheet

ADCPSSI – Processor sample sequence initiate



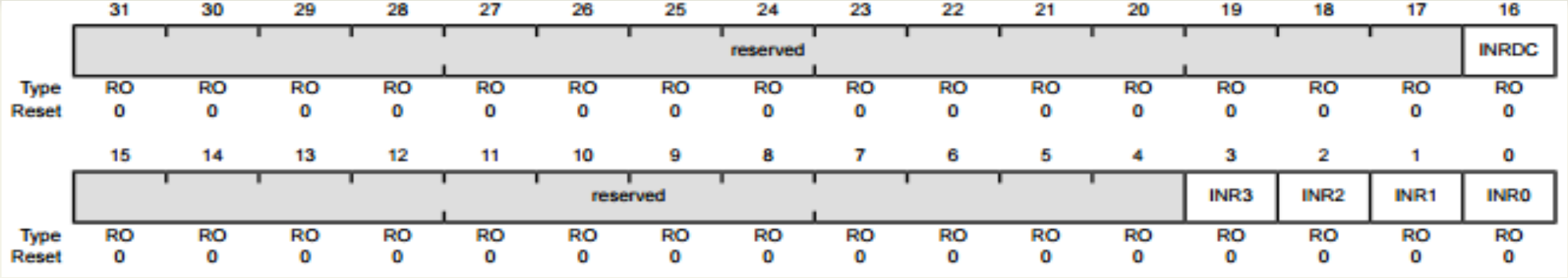
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GSYNC	reserved			SYNCWAIT	reserved										
Type	RW	RO	RO	RO	RW	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												SS3	SS2	SS1	SS0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-

3 SS3 – SS3 initiate, 1 begins sampling SS3
2 SS2 – SS2 initiate, 1 begins sampling SS2
1 SS1 – SS1 initiate, 1 begins sampling SS1
0 SS0 – SS0 initiate, 1 begins sampling SS0

This is used to trigger the start of a conversion.

ADCRIS – Raw interrupt status



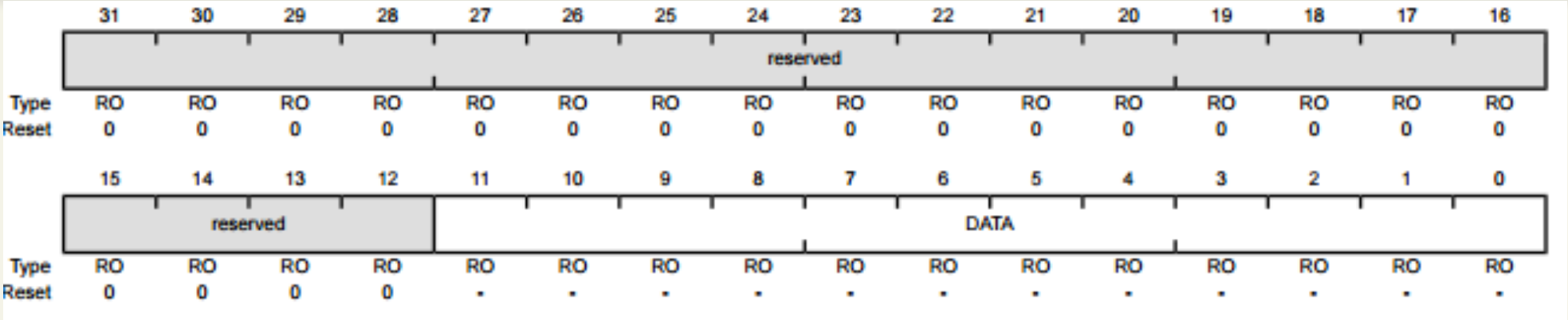
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															INRDC
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												INR3	INR2	INR1	INR0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3 INR3 – SS3 raw interrupt status, 1 means conversion complete on sample sequencer
2 INR2 – SS2 raw interrupt status, 1 means conversion complete on sample sequencer
1 INR1 – SS1 raw interrupt status, 1 means conversion complete on sample sequencer
0 INR0 – SS0 raw interrupt status, 1 means conversion complete on sample sequencer

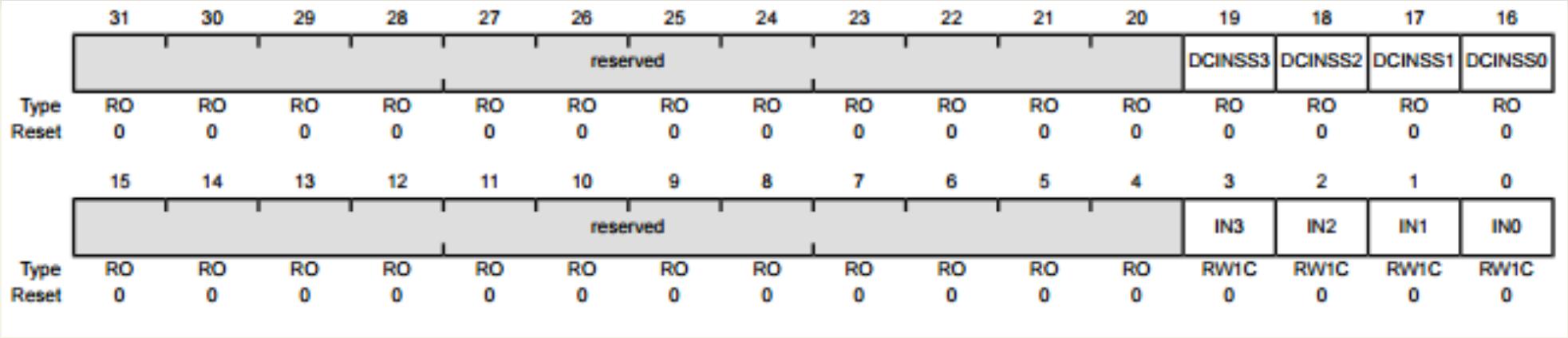
In lab we can use this to check when conversions are complete.

ADCSSFIFOn – Sample sequence result FIFO



11:0 DATA – contains the conversion result for SSn

ADCISC – Interrupt status and clear



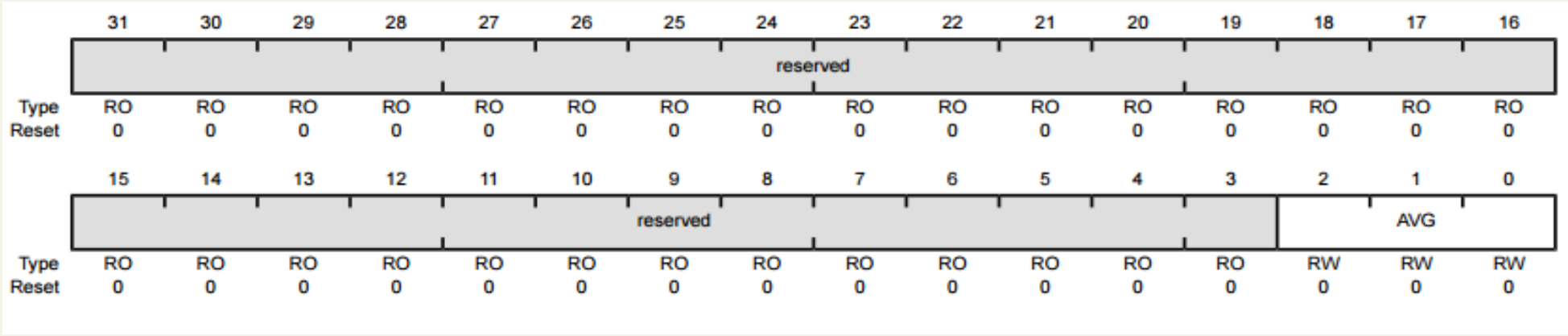
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved												DCINSS3	DCINSS2	DCINSS1	DCINSS0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved												IN3	IN2	IN1	IN0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW1C	RW1C	RW1C	RW1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3:0 INn – write 1 to clear flag in ADCRIS

In lab we will only be using INn bits.

ADCSAC – Sample averaging control



	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved													AVG		
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2:0 AVG –

- 0x0 = No oversample
- 0x1 = 2x hardware oversample
- 0x2 = 4x hardware oversample
- 0x3 = 8x hardware oversample
- 0x4 = 16x hardware oversample
- 0x5 = 32x hardware oversample
- 0x6 = 64x hardware oversample
- 0x7 = reserved

ADC Register Usage List

- ADC Run Mode Clock Gating Control (**RCGCADC**): Control the clocking to enable the ADC module
- ADC Peripheral Ready (**PRADC**): Check whether the ADC module is ready for access
- ADC Active Sample Sequencer (**ADCACTSS**): Enable or disable a sample sequencer (SS) unit
- ADC Processor Sample Sequence Initiate (**ADCPSSI**): Initiate a sample in a SS (start ADC conversion)
- ADC Raw Interrupt Status (**ADCRIS**): Flag to indicate whether ADC conversion is done in SS (may be polled by software without sending an interrupt to the NVIC), status bit for each SS
- ADC Interrupt Status and Clear (**ADCISC**): Clear SS done status by writing 1

ADC Register Usage List (continued)

- ADC Sample Sequence Result FIFO (0-3) (**ADCSSFIFO0-3**): Contains conversion result(s) for samples taken by an SS (one SSFIFO register for each SS), consecutive reads return samples 0, 1, etc.
- ADC Event Multiplexer Select (**ADCEMUX**): Select the event that initiates a sample for each SS, use default of 0 (processor, which means software via PSSI)
- ADC Sample Sequence Input Multiplexer Select (0-3) (**ADCSSMUX0-3**): Select the analog input channel for each sample taken by the SS (one SSMUX register for each SS, 4-bit field for each sample, sequence starts with sample 0)
- ADC Sample Sequence Control (0-3) (**ADCSSCTL0-3**): For each sample, configure the sample in 4 bits (e.g., 0b0110 means stopping with this sample and raising the RIS flag when done with the sample)