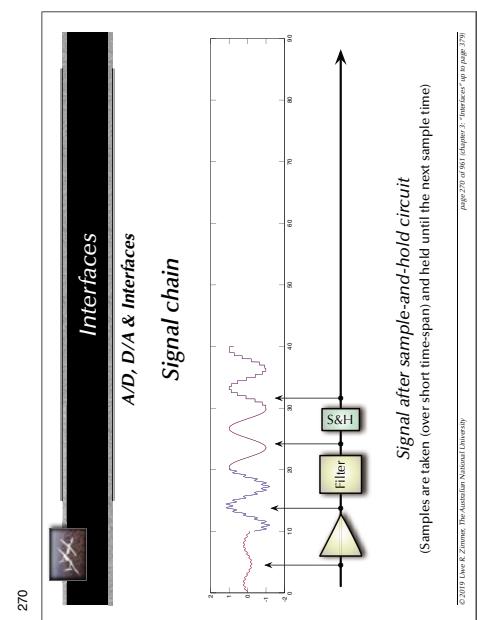
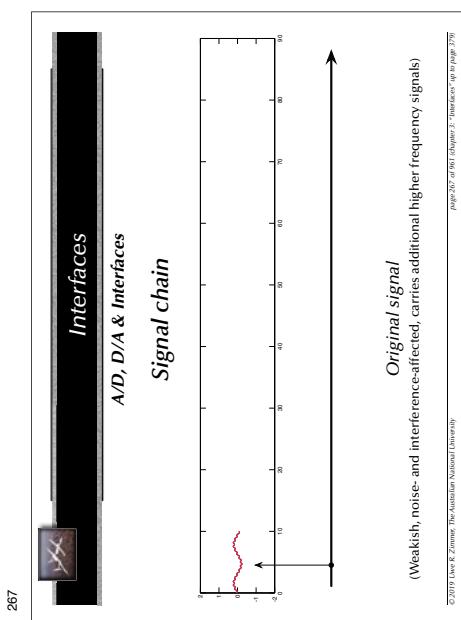


**Real-Time & Embedded Systems 2019**

**3**

Interfaces  
Uwe R. Zimmer - The Australian National University

© 2019 Uwe R. Zimmer, The Australian National University

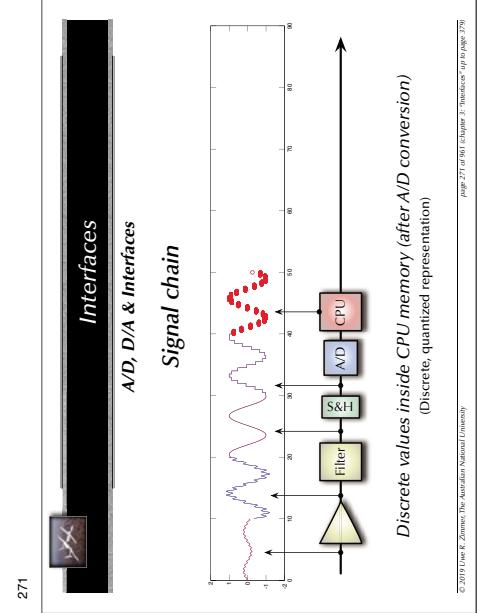
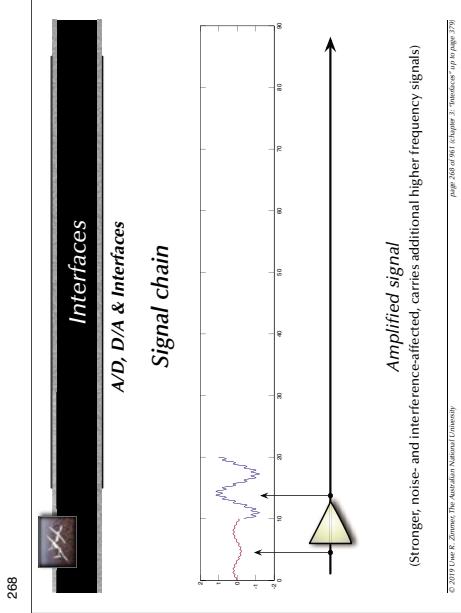


**Interfaces**

**References**

- [Burns2007] Burns, A. & Wellings, A. *Concurrent and Real-time Programming in Ada, edition 1*. Cambridge University Press, 2007
- [Motorola1996] Motorola. *Time Processing Unit Reference Manual* 1996 pp. 1-142
- [Motorola2000] Motorola. *ADC 08200 - 8 bit, 20 MSPS* Datasheet 2002 pp. 1-19
- [Peacock1997] Peacock, GR. *Standards for temperature sensors*. <http://www.temperatures.com/resources/standards/>, 1997
- [Semiconductors1999] National Semiconductors. *LM 724/58 - 12 bit, Data Acquisition* 1999 pp. 1-36
- [Semiconductors2002] National Semiconductors. *ADC 08200 - 8 bit, 20 MSPS to 200 MSPS* Datasheet 2002 pp. 1-19

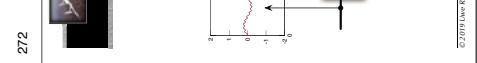
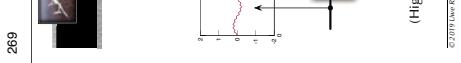
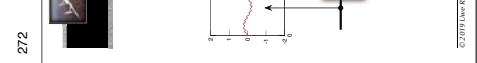
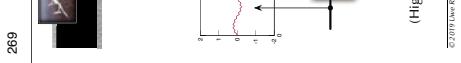
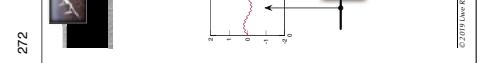
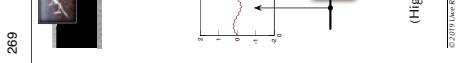
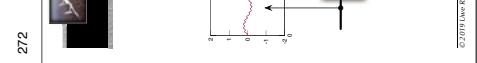
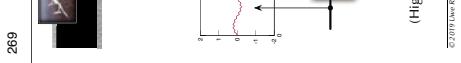
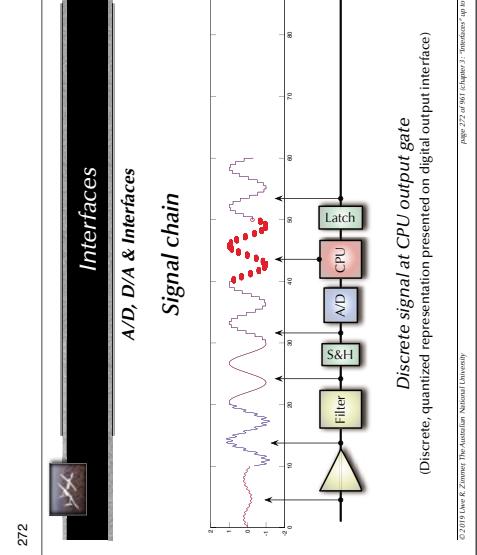
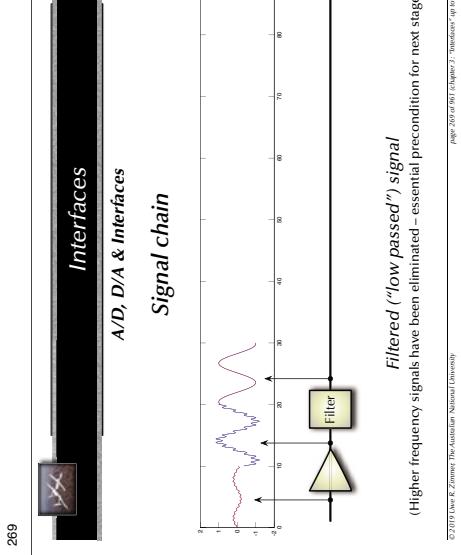
© 2019 Uwe R. Zimmer, The Australian National University



**Interfaces**

**Real-Time Systems Components: Interfaces**

© 2019 Uwe R. Zimmer, The Australian National University



**Interfaces**

**A/D, D/A & Interfaces**

**Signal chain**

**Analogue signal after D/A conversion**  
(Limited bandwidth leads to glitches in the analogue signal)

Page 273 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Sampling**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

Page 276 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Nyquist's Criterion**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

The phenomenon of the wrongly observed signal at a lower frequency  $f_s$  is called aliasing.

Page 279 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Signal chain**

**Smoothed ("deglitched") signal**  
(Synchronized filter leads to predictable, analogue transition steps)

Page 274 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Sampling**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

Page 277 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Nyquist's Criterion**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

An analog signal with bandwidth  $f_a$  must be sampled at:  $f_s > 2f_a$

Perfect measurements taken at  $f_s > 2f_a$  result in no information loss due to sampling.

Due to actual (quantized) measurements; oversampling is required.

Page 278 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Signal chain**

**Filtered ("low passed") signal**  
(Signals introduced by the conversion process are eliminated)

Page 275 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Sampling**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

Page 279 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

**Nyquist's Criterion**

**Sample data with frequency  $f_s$**   
Interpolation suggests a source signal .....

An analog signal with bandwidth  $f_a$  must be sampled at:  $f_s > 2f_a$

Perfect measurements taken at  $f_s > 2f_a$  result in no information loss due to sampling.

Due to actual (quantized) measurements; oversampling is required.

Page 280 of 961 Chapter 2: Interfaces up to page 279  
© 2019 Univ. K. Zinner, The Australian National University

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

A resolution of  $N$  bits provides  $2^N$  possible discrete output levels:

- Smallest distinguishable value  $q$  (**Least Significant Bit or LSB**):  

$$q = \frac{1}{2^N}$$
- Ratio  $\frac{1}{q}$  expressed in decibel (dB):  

$$10\log 2^{2N} = N \cdot 20\log 2 \approx N \cdot 6.02\text{dB}$$
  
 (Decibel (dB) is a ratio of powers defined as:  $10\log \frac{P_1}{P_0}$  or by signal amplitudes:  $10\log \frac{A_1^2}{A_0^2}$ )

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 283 of 961 (Chapter 3: Interfaces up to page 279)

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)

Page 287 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

Assuming an ideal input signal:  
 $F(t) = A \sin \omega t$  with  $q = \frac{2A}{2^N}$  and  $S^2 = \frac{A^2}{2}$   
 then the signal to noise ratio is:  
 $SNR [dB] = 10\log\left(\frac{S^2}{q^2}\right) = 10\log\left(\frac{3 \cdot 2^{2N}}{2}\right)$   
 $SNR [dB] \approx 20N\log 2 + 10\log\frac{3}{2}$   
 $SNR [dB] \approx N \cdot 6.02 + 1.76$

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)

Page 287 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### A/D converters

Some criteria to select the fitting A/D converter for an application:

- Throughput (maximal sampling frequency).
- Accuracy (ENOB, SNR).
- Latency (time from sensing to delivery).
- Power consumption.
- Complexity (also affects: price).

**Trade-offs are to be expected:**

- Maximizing throughput will reduce accuracy and increase power consumption.
- Maximizing accuracy will increase latency ... other trade-offs

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 288 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

The mean square error over one step:

$$\overline{E^2} = \frac{1}{q} \int q E^2 dE = \frac{q^2}{12}$$

**Root mean square (rms) noise voltage:**

$$\frac{q}{\sqrt{12}}$$

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)

Page 287 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

Determining the effective number of bits (ENOB):

$$SNR_{ideal} [dB] = 20N\log 2 + 10\log\frac{3}{2}$$

$$ENOB = \frac{SNR_{actual} - 10\log\frac{3}{2}}{20\log 2}$$

$$ENOB = \frac{SNR_{actual} - 1.76}{6.02}$$

$$ENOB = N \text{ for } SNR_{actual} = SNR_{ideal}$$

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)

Page 287 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Integrating A/D converters (Single Slope)

Integrate a reference voltage  $U_{ref}$  until it matches the input voltage  $A_{IN}$ :

- Sampling frequency depends on input signal.
- Accuracy depends on  $U_{ref}$ , integrator and clock.
- Simple components.
- Slow (typically  $\sim 100\text{Hz}$ )

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 288 of 961 (Chapter 3: Interfaces up to page 279)

Page 289 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

The signal  $S$  with respect to the rms noise:

$$\frac{S}{\sqrt{q}} = S \cdot 2^N \sqrt{\frac{1}{12}}$$

or as the **signal to noise ratio in decibel**:

$$SNR [\text{dB}] = 10 \log\left(\frac{S^2}{q^2}\right)$$

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 284 of 961 (Chapter 3: Interfaces up to page 279)

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)

**Interfaces**

**A/D, D/A & Interfaces**

### Quantization

Actual A/D converters are also characterized by:

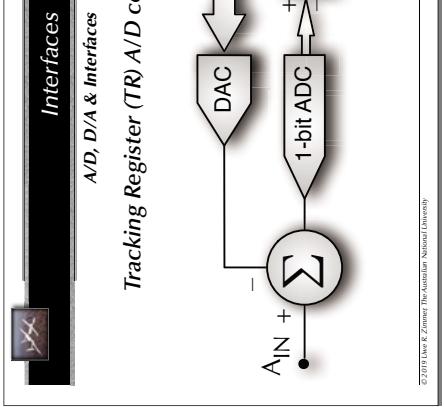
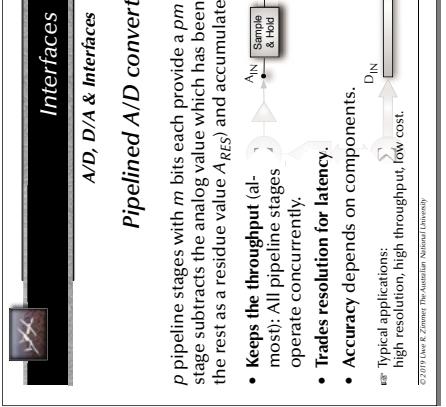
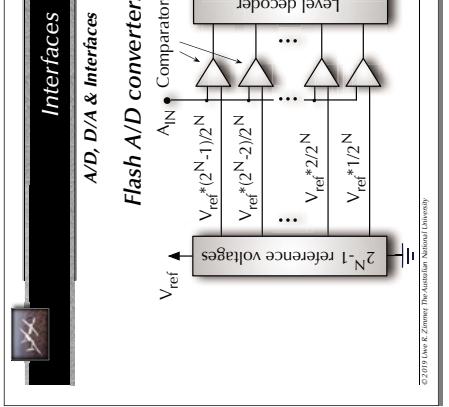
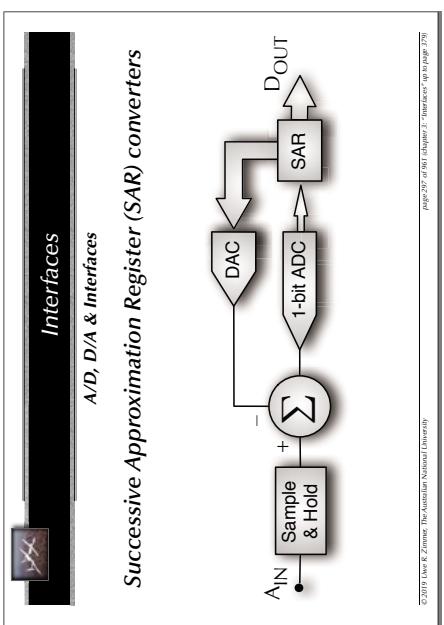
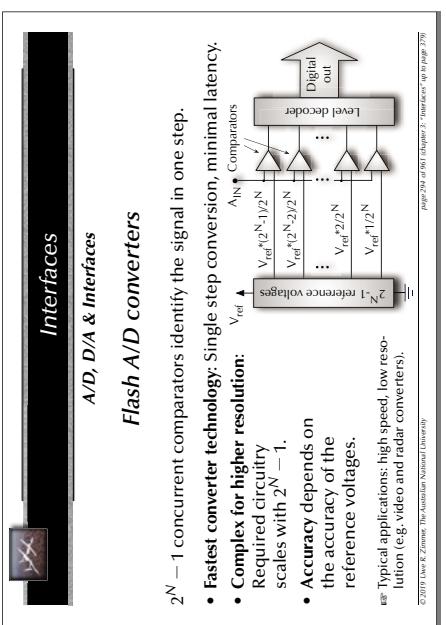
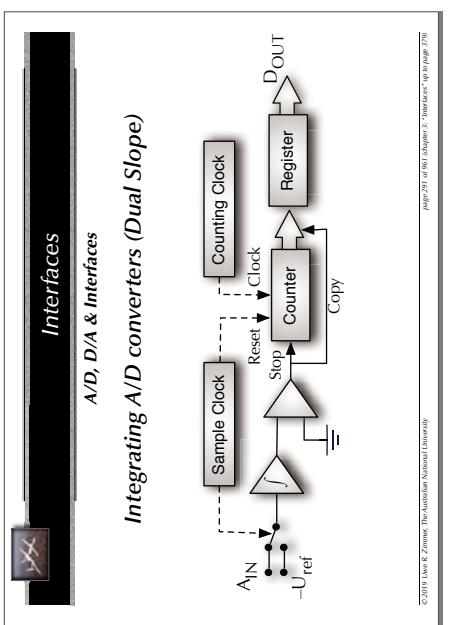
- Integral Non-Linearity (INL):** Maximal difference between the actual and ideal code centres.
- Differential Non-Linearity (DNL):** Differences between successive code widths.
- Missing codes:** reduce SNR by  $20\log 2$  or  $6.02\text{dB}$  for each missing code.
- Response time / Latency, Throughput / Maximal sampling rate**

© 2019 Dr. Uwe R. Zimmer, The Australian National University

Page 284 of 961 (Chapter 3: Interfaces up to page 279)

Page 285 of 961 (Chapter 3: Interfaces up to page 279)

Page 286 of 961 (Chapter 3: Interfaces up to page 279)



**Interfaces**

**A/D, D/A & Interfaces**

### Tracking Register (TR) A/D converters

Continuous single bit conversion compares the current digital output with the analog input and counts a register up/down accordingly.

- Minimal circuitry (no S&H) – (almost) independent of resolution.
- Speed depends on amplitude changes in the input signal  $\Leftrightarrow$  no constant sampling frequency.
- Accuracy depends on the accuracy of the single bit ADC and the DAC.

$\Rightarrow$  Typical applications: Tracking slowly changing signals at high frequency. Rarely used today.

© 2019 Iain K. Zettler, The Australian National University  
Page 301 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

**$\Sigma$ - $\Delta$  A/D converters**

The effective number of bits (SNR) is given by  $\text{SNR} = 6.02 \times \text{bits} + 1.3$ .

© 2019 Iain K. Zettler, The Australian National University  
Page 302 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### Higher order $\Sigma$ - $\Delta$ A/D converters

Effective Number Of Bits (Signal to Noise Ratio) can be improved by adding further integrator stages.

**2nd order  $\Sigma$ - $\Delta$**

$\Rightarrow$  SNR is proportional to  $\text{SNR} = 6.02 \times \text{bits} + 1.3$ .

**•  $\Sigma$ - $\Delta$  converters are not subject to aliasing, as they implicitly implement a low pass filter via the integrators.**

© 2019 Iain K. Zettler, The Australian National University  
Page 306 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ A/D converters

$\Rightarrow$  Typical applications: Tracking slowly changing signals at high frequency. Rarely used today.

© 2019 Iain K. Zettler, The Australian National University  
Page 303 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ A/D converters

The sampling frequency of the bitstream with respect to the digital output frequency (oversampling) determines accuracy.

- Latency depends on bitstream frequency.
- Accuracy depends on number of bits decimated.
- Method is inherently linear.

$\Rightarrow$  Typical applications: High accuracy (typically 16-24 bits), moderate throughput (24bit high quality audio converters are usually of this type).

© 2019 Iain K. Zettler, The Australian National University  
Page 304 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ D/D converters

Digital form converts to bitstream or to analog.

$\Rightarrow$  Typical application: Audio bitstreams.

© 2019 Iain K. Zettler, The Australian National University  
Page 305 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ A/D converters

$+ U_{\text{ref}}$  or  $- U_{\text{ref}}$  is subtracted from the input signal and integrated.

The high frequent comparison of the integrator against ground results in a bitstream signal (which is also fed back). The density of '1's in the bitstream represents the input signal.

The bitstream can be deployed as such or be translated into digital words of varying lengths.

© 2019 Iain K. Zettler, The Australian National University  
Page 306 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ D/D converters

$\Rightarrow$  Typical application: Audio bitstreams.

© 2019 Iain K. Zettler, The Australian National University  
Page 307 of 360 | Chapter 3: Interfaces | up to page 370

**Interfaces**

**A/D, D/A & Interfaces**

### $\Sigma$ - $\Delta$ converters matrix

	Integrating	SAR	$\Sigma$ - $\Delta$	Pipeline Flash	Flash
Throughput	$O(\frac{1}{2^k})$	$O(\frac{1}{2^k})$	throughput vs. latency	$O(1)$	$O(1)$
Latency	$O(2^k)$	$O(N)$	latency vs. accuracy	$O(p)$	$O(1)$
Accuracy	can be high	medium-high	typ. high accuracy	typ. low-medium	typ. low
Resolution	typ. 8-16bit	typ. 8-24bit	typ. 8-16bit	typ. 8-16bit	typ. 4-8bit
Size / Components	$O(1)$	$O(1)$	$O(1)$	$O(p \cdot 2^k)$	$O(2^k)$
Notes	Can support some frequencies	Cost efficient and can be very accurate	Flexible architecture, compromise between speed and cost	Faster converter	

© 2019 Iain K. Zettler, The Australian National University  
Page 308 of 360 | Chapter 3: Interfaces | up to page 370







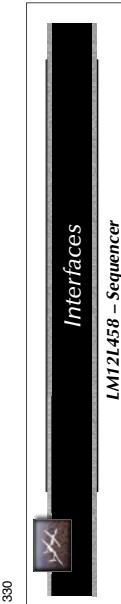
### LM12L458 – Instructions

```

IP := 0;
Loop
repeat
    if Auto_Zero or Full_Cal then Calibrate;
    until bit;
    if Instr(IP).Watchdog then Wait_for_external_sync;
    Current_Signal := Aquisition_Instr(IP).Aquisition_Time;
    if Instr(IP).Timer then Run_Timer;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_1;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_2;
    else
        if Instr(IP).Sync then Wait_for_external_sync;
        Convert_and_Store_INTO (current_Signal);
    end;
    if Instr(IP).Loop then IP := 0;
    else IP := IP + 1;
end Loop;

```

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



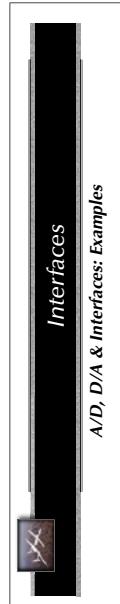
### LM12L458 – Sequencer

```

IP := 0;
Loop
repeat
    if Auto_Zero or Full_Cal then Calibrate;
    until bit;
    if Instr(IP).Watchdog then Wait_for_external_sync;
    Current_Signal := Aquisition_Instr(IP).Aquisition_Time;
    if Instr(IP).Timer then Run_Timer;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_1;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_2;
    else
        if Instr(IP).Sync then Wait_for_external_sync;
        Convert_and_Store_INTO (current_Signal);
    end;
    if Instr(IP).Loop then IP := 0;
    else IP := IP + 1;
end Loop;

```

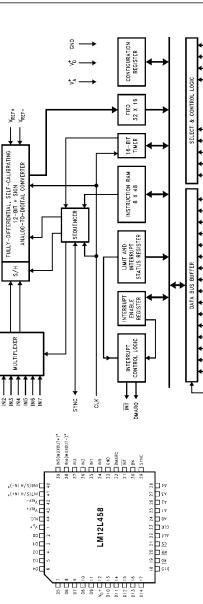
© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



### A/D, D/A & Interfaces: Examples

#### LM12L458

(National Semiconductors)



© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



### LM12L458 – Configuration register

- Start (1bit): starts the sequencer.
- Reset (1bit): sets the instruction pointer to '000'.
- Auto\_Zero (1bit): triggers a 'short calibration' 76 cycles -1 offset sample.
- Full\_Cal (1bit): initiates a full calibration (4944 cycles, 8 samples) for testing purposes.
- Standby (1bit): disconnects the external clock and preserves the registers.
- After powering up again (~10ms), a specific interrupt is issued.
- Chan\_Mask (1bit): formal selection for the FIFO output registers.

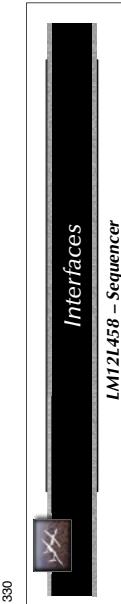
Configuration register entries consist of:

- Start (1bit): sets the instruction pointer to '000'.
- Reset (1bit): sets the sync pin to input or output mode.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



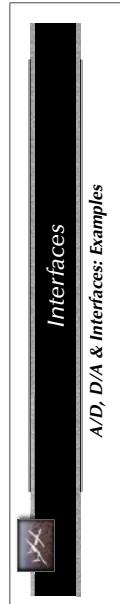
### LM12L458 – Sequencer

```

IP := 0;
Loop
repeat
    if Auto_Zero or Full_Cal then Calibrate;
    until bit;
    if Instr(IP).Watchdog then Wait_for_external_sync;
    Current_Signal := Aquisition_Instr(IP).Aquisition_Time;
    if Instr(IP).Timer then Run_Timer;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_1;
    if Instr(IP).Sync then Wait_for_external_sync;
    Compare_Current_Instr(IP).Limit_2;
    else
        if Instr(IP).Sync then Wait_for_external_sync;
        Convert_and_Store_INTO (current_Signal);
    end;
    if Instr(IP).Loop then IP := 0;
    else IP := IP + 1;
end Loop;

```

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



### A/D, D/A & Interfaces: Examples

#### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



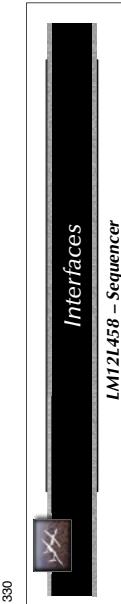
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



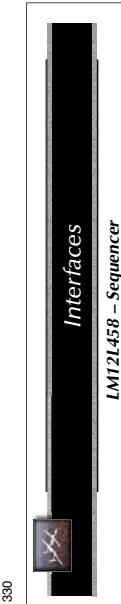
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



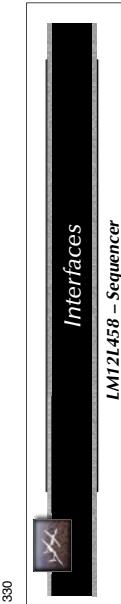
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



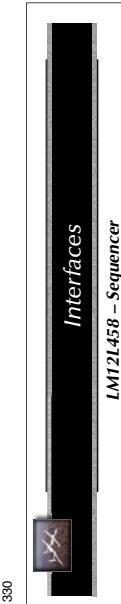
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



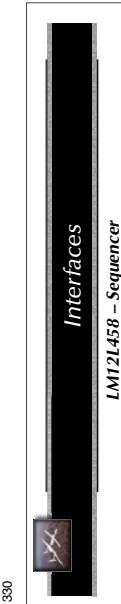
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```



### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1 0 0 0	Configuration	RW	Registor															
0 1 0 1	Acquisition	RW	Registor															
1 1 1 1	Power	0x00	Registor															

© 2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)



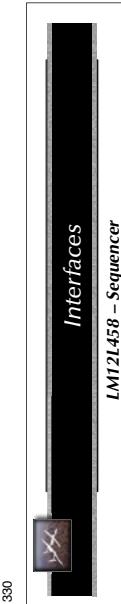
### LM12L458 – Configuration register

- Auto\_Zero= (1bit): auto-zeros the ADC automatically in every conversion.
- I/O Sel (1bit): sets the Sync pin to input or output mode.
- RAM Pointer (2 bits): selects the current (16-bit) part in each 48-bit instruction.
- Test=0 (1bit): production testing mode; leave this bit at '0'.
- DIAG (1bit): connects  $V_{IN+}$  and  $V_{IN-}$  to  $V_{REF+}$  and  $V_{REF-}$  for testing purposes.

```

{2019 Use R. Zimman, The Australian National University  
Page 331 of 951 (Chapter 5: Interfaces up to page 379)
}

```

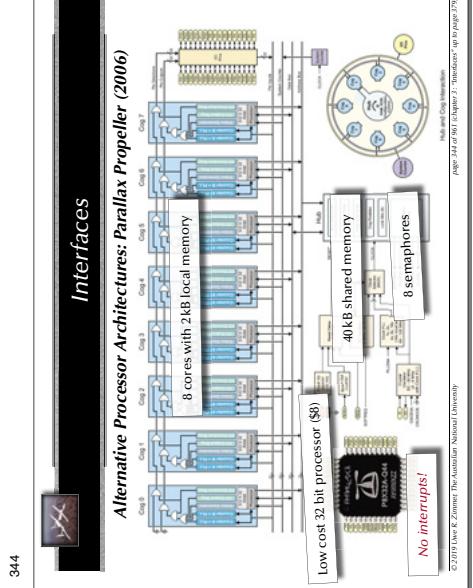
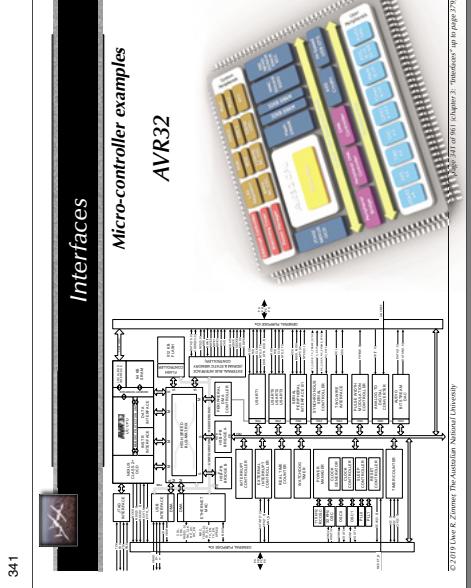
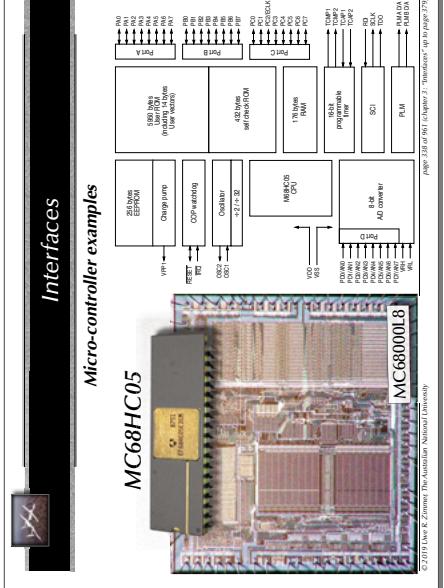
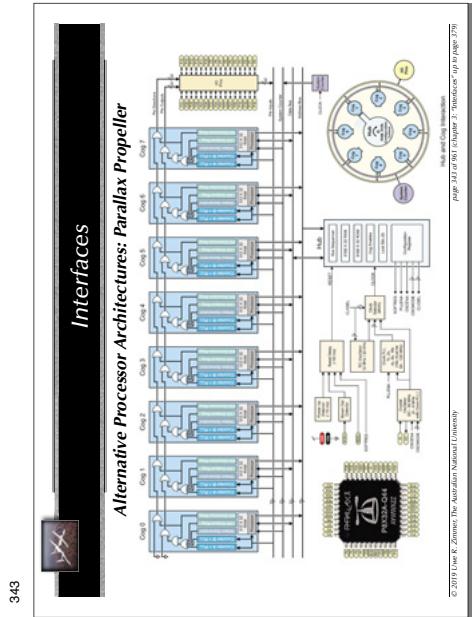
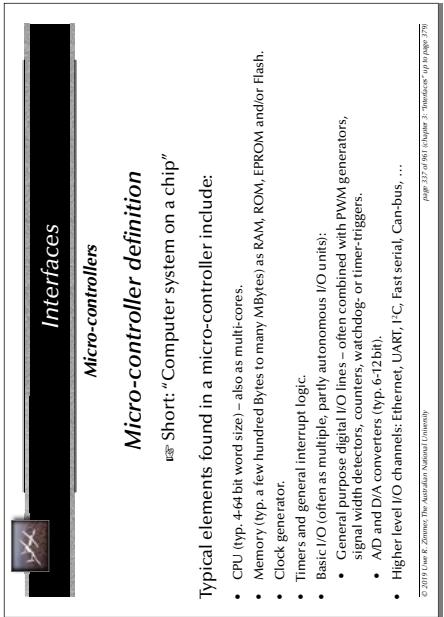
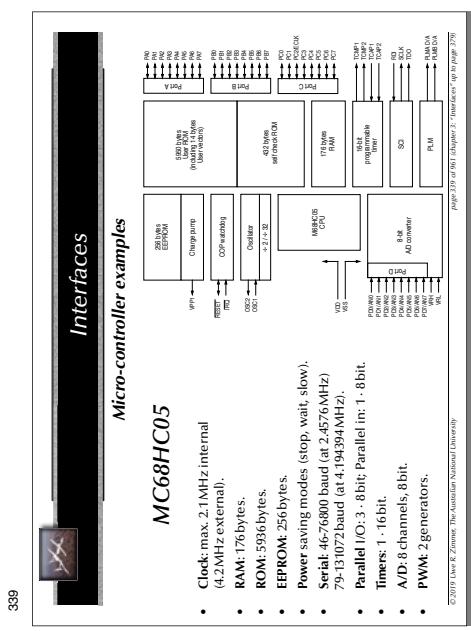
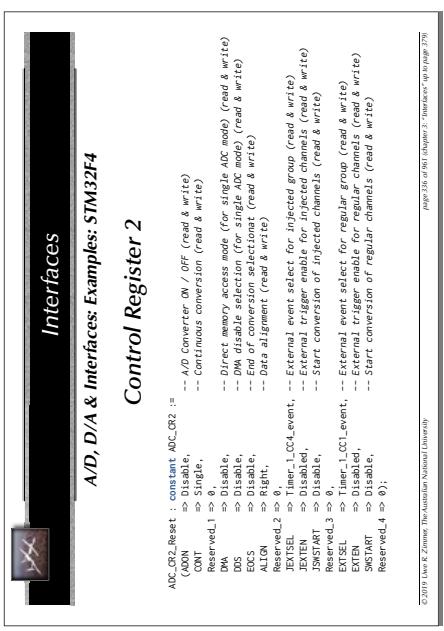


### LM12L458 – Configuration register

AS AS A1	Purpose	Type	Disp	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0




<tbl\_r cells="19" ix="4" maxcspan="1" max





## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	

357

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

358

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

359

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

360

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

361

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

362

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

363

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	
4: Jump, flag, and RAM:	
5: Execution unit, immediate, and flag:	

364

## Interfaces

### Micro-controller examples

#### Time Processing Unit – μinstructions formats:

2: Execution unit, flag, and channel control:	
5: Execution unit, immediate, and flag:	<img alt="Format diagram for execution unit, immediate, and flag. It shows fields: Y[31], X[30], S[29], T[28], D[27], R[26], SH[25], C[24

**Interfaces**

**Micro-controller examples**

### TPU Fixed scheduled, prioritized time slots

**Determining actual TPU latencies**

States have different and variable lengths.  
☞ Calculating actual and maximal latencies requires full understanding of all states.

Page 363 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

Page 364 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

366

**Interfaces**

**Micro-controller examples**

### Determining actual TPU latencies

**Determining actual TPU latencies**

Emulate the known memory access patterns for all states.  
☞ Add two 2 µ-cycles for each memory access (potential stall times).

Page 365 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

Page 366 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

367

**Interfaces**

**Micro-controller examples**

### Determining maximal TPU latencies

**Determining maximal TPU latencies**

Emulate all states on the same priority runnable at all times and run their maximal lengths.  
☞ Assume the longest state out of all higher priorities runnable at all times.  
☞ Assume the longest states on each lower priority runnable at all times.  
Deploy a set out of those states which will cause the longest latency.  
☞ Determine the longest latency inside a full hyper-cycle.

Page 367 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

Page 368 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

368

**Interfaces**

**Micro-controller examples**

### Latencies on TPUs

**Determining actual TPU latencies**

- ... for latencies of capture and match at each channel mind:
- only the time-base resolution (all channels are evaluated independently and in parallel).

... for the functions associated with individual channels mind the:

- number of active channels (max. 16).
- number of channels on each priority level (add max. 2 µ-cycles for each "state-switch").
- number of available time slots on each priority level per full scheduler-cycle (4, 2, 1 slot(s)).
- number of µ-cycles to execute individual states of a function (2 µ-cycles per µ-instruction).
- number of RAM accesses during the execution of a state  
(each access may stall for 2 CPU-µcycles).
- TPU clock cycle frequency.

Page 364 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

366

**Interfaces**

**Micro-controller examples**

### Determining maximal TPU latencies

**Determining maximal TPU latencies**

Assume all states on the same priority runnable at all times and run their maximal lengths.  
☞ Assume the longest state out of all higher priorities runnable at all times.  
☞ Assume the longest states on each lower priority runnable at all times.  
Deploy a set out of those states which will cause the longest latency.  
☞ Determine the longest latency inside a full hyper-cycle.

Page 365 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

Page 366 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

367

**Interfaces**

**Micro-controller examples**

### Time Processing Unit

A special purpose micro-controller:

- Independent µ-engine.
- 16 digital I/O channels with independent match and capture capabilities.
- Mean to operate these I/O channels for timing control purposes.
- Predefined µ-engine command set (ROM functions in control store),
- 2 16-bit time bases

**64 bit time base**  
(driven by an external clock; e.g. 20MHz; resolution: 50ns; range: -30,000 years).

**Free running**  
(not influenced by any CPU action or resets).

**2 reference registers**  
(used for compares and interrupt generation).

**Real-Time clock**  
supplies full seconds (32bit<sup>8</sup>; range: -136 years)  
(not affected by CPU, resets, and operates in all low-power modes).

Page 371 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

Page 372 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

368

**Interfaces**

**Micro-controller examples**

### Determining actual TPU latencies

**Determining actual TPU latencies**

Emulate the known executing ready times for all states.  
☞ Add two 2 µ-cycles for each state switch.

Page 365 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

366

**Interfaces**

**Micro-controller examples**

### Determining maximal TPU latencies

**Determining maximal TPU latencies**

Assume all states on the same priority runnable at all times and run their maximal lengths.  
☞ Assume the longest state out of all higher priorities runnable at all times.  
☞ Assume the longest states on each lower priority runnable at all times.  
Deploy a set out of those states which will cause the longest latency.  
☞ Determine the longest latency inside a full hyper-cycle.

Page 366 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

367

**Interfaces**

**Micro-controller examples**

### MPC565 : Time-base & Real-time clock

**64 bit time base**  
(driven by an external clock; e.g. 20MHz; resolution: 50ns; range: -30,000 years).

**Free running**  
(not influenced by any CPU action or resets).

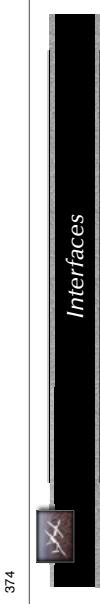
**2 reference registers**  
(used for compares and interrupt generation).

**Real-Time clock**  
supplies full seconds (32bit<sup>8</sup>; range: -136 years)  
(not affected by CPU, resets, and operates in all low-power modes).

Page 373 of 561 (Chapter 3: Timebases" up to page 370)

© 2019 Iain R. Zugner, The Australian National University

368



## MPC565 : Interrupt controller

### Micro-controller examples

### MPC565 : Nexus debug port (IEEE-1588-1999)

(Real-time embedded application development interface)

#### On-Line mode:

- Program trace: via branch trace messaging.
- Data trace: via data writer messaging and data read messaging (can be reduced to selected areas).
- Owner trace: via ownership trace messaging (also indicates task creation and activation).
- Run-time access to memory map and special CPU registers.
- Watchpoints: CPU watchpoint status signals are snooped and transferred with high priority.

#### Off-line mode:

- Read / Write access: the READY module can take over the I-bus to manipulate data.
- Access to all CPU registers during halt.

© 2010 University of Zaragoza, The Australian National University

375

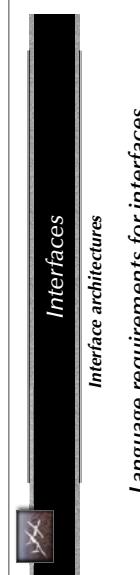


## Basic sampling control mechanisms

- Status driven:** the computer polls for information (used in dedicated micro-controllers and pre-scheduled hard real-time environments).
- Interrupt driven:** The data generating device may issue an interrupt when new data has been detected / converted or when internal buffers are full.
- Program controlled:** The interrupts are handled by the CPU directly, by changing tasks, calling a procedure, raising an exception, free tasks on a semaphore, sending a message to a task, ...).
- Program initiated:** The interrupt is handled by a DMA-controller. No processing is performed. Depending on the DMA setup, cycle stealing can occur and needs to be considered for the worst case computing times.
- Channel program controlled:** The interrupts are handled by a dedicated channel device. The data is transferred and processed. Optional memory-based communication with the CPU. The channel controller is usually itself a dedicated µ-engine / -controller.

© 2010 University of Zaragoza, The Australian National University

376



## Language requirements for interfaces

- Specify the device interface (protocol and formats) in all detail**  
(candidates: Ada, CHILL, ERLANG, Modula-2, C, ...)

...or Macro-Assemblers level (if platform-independence or abstraction is not required).

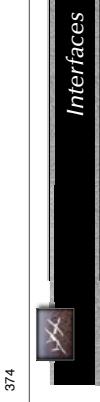
- Handling asynchronous hardware messages (devices, timers, ...)**  
Many different methods to implement a context-switch  
(candidates: all languages with some real-time orientation:  
PEARL, CHILL, ERLANG, Ada, RT-Java, POSIX, ...)

...or the term "high-level languages" in the real-time interface context:

- Allow for  
**strong abstractions while being time and physics specific**  
down to the actual level of interface realities

© 2010 University of Zaragoza, The Australian National University

378



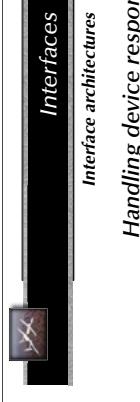
## MPC565

(Power: power dissipation: 0.6-1.12 W, -40° +125°C

- CPU PowerPC core (incl. FPU & BBC), 50MHz
- RAM: flash: 1MB static, 36 kB
- Time processing units: 3 (via dual-ported RAM)
- Timers: 22 channels (PWM & RTC supported)
- A/D converters: 40 channels, 10bit, 250kHz
- Can-bus: 3 CAN modules
- Serial: 2 interfaces
- Data link controller:  
SAE 1850 class B communications module
- Real-time embedded application development interface: NEXUS debug port (IEEE-1588-1999)
- Parking: 352/388 ball PBGA

© 2010 University of Zaragoza, The Australian National University

377



## Interface architectures

### Handling device responses

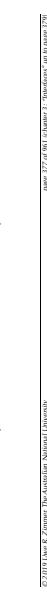
↳ How to embed the unpredictable in predictable systems?

By providing the resources to cope with the assumed worst case  
(and fall back to a lower, yet safe functionality beyond that).

Concrete:

- Either the unpredictable events need to be synchronized with the remaining real-time tasks without violating real-time constraints,
- Or exclusive processing resources (e.g. a dedicated micro-controller) for a specific device need to be provided.

© 2010 University of Zaragoza, The Australian National University



## Converters & Interfaces

### Summary

- Analogue signal chain in a digital system**
  - Sampling data, aliasing, Nyquist's criterion, oversampling
  - Quantization (LSB, rms noise voltage, SNR, ENOB), Missing codes, DNL, INL

#### A/D converters:

- Integrating (Single- / Dual-slope), Flash, Pipelined, SAR, Tracking, ΣΔ, ΣΔA, DDA, n-th order ΣΔ.
- Examples:**
  - Fast and simple A/D converter example: National Semiconductor ADC08020
  - Multi-channel A/D data logging interface example: National Semiconductor LM121458
  - Simple 8-bit µ-controller example: Motorola MC68HC05, Propeller.
  - Complex 32-bit µ-controller examples: AVR32 and Motorola MPC565 (including TPUs).

- General device handling / sampling control / language requirements**
  - Strong abstractions while being time and physics specific
  - down to the actual level of interface realities

© 2010 University of Zaragoza, The Australian National University

379