eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Teknologia Elektronikoa Saila

KONPUTAGAILUEN ARKITEKTURA
80c552 - ADC

Kudeaketa eta Informazio Sistemen Informatikaren Ingenieritzako Graduaren 3. maila

Irakaslea: Alain Sanchez
(alain.sanchez@ehu.eus)

2019-2020, 1. lauhilabetea

HELBURUAK

Aldagai analogikoekin zelan lan egin daitekeen ulertzea

80c552-aren ADC-a zelan konfiguratzen den ikastea

ADC

Ze aldagai mota egon daitezke? Zer da ADC bat?

ADC

Ze aldagai mota egon daitezke? Zer da ADC bat?

Zelan lan egiten da aldagai analogikoekin?

ADC

Ze aldagai mota egon daitezke? Zer da ADC bat?

Zelan lan egiten da aldagai analogikoekin?

Ze balioraino neurtu dezakegu? Ze resoluzioarekin? Denboran? Anplitudean?

ADC

ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

ADC

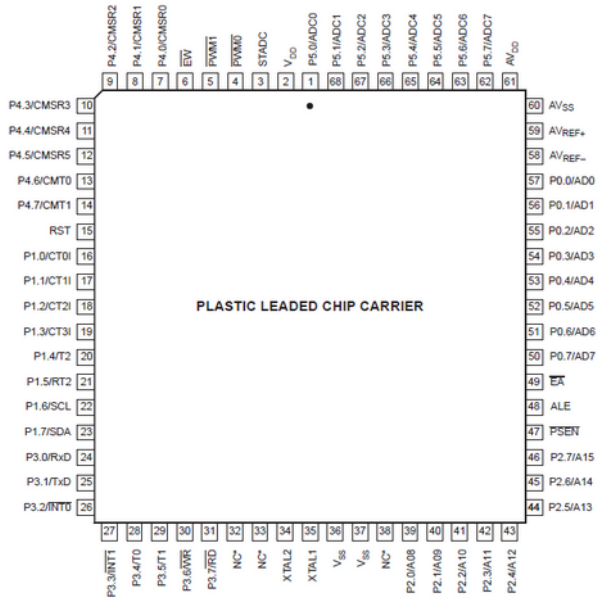
ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

P5 portua erabili. Beraz, 8 ADC kanal ezberdin.

Port 5 Operation

Port 5 may be used to input up to 8 analog signals to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input logic from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (Ct) should be taken into consideration when both analog and digital signals are simultaneously input to Port 5 (see, D.C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in Table 10. A more detailed description of these features can be found in the relevant parts of this section.



ADC

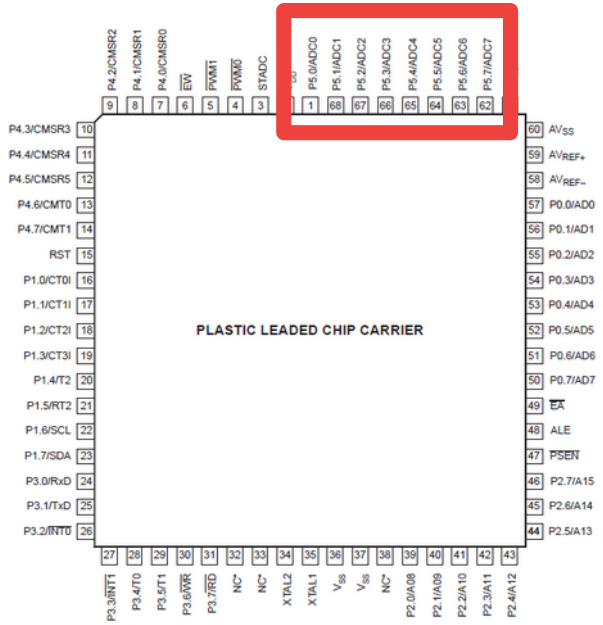
ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

P5 portua erabili. Beraz, 8 ADC kanal ezberdin. Multipex.

Port 5 Operation

Port 5 may be used to input up to 8 analog signals to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input logic from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (Ct) should be taken into consideration when both analog and digital signals are simultaneously input to Port 5 (see, D.C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in Table 10. A more detailed description of these features can be found in the relevant parts of this section.

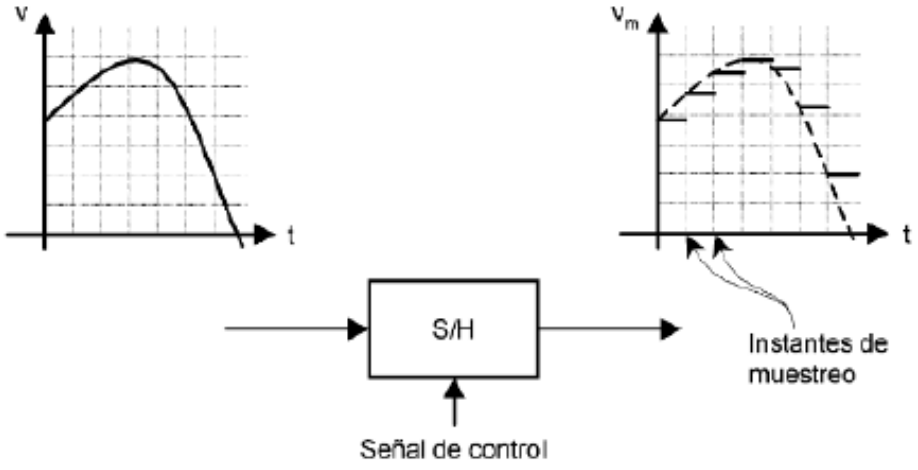


ADC

ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

P5 portua erabili. Beraz, 8 ADC kanal ezberdin. Multipex.

Diskretua bai denboran, eta baita anplitudean ere.



ADC

ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

P5 portua erabili. Beraz, 8 ADC kanal ezberdin. Multipex.

Diskretua bai denboran, eta baita anplitudean ere.

10 bit portu bakoitzeko. (2^(10))-eko resoluzioa. Tentsioan, Vcc/(2^(10))

ADC

ADC: Analog to Digital Converter. Hau da, aldagai analogiko bat, digital bihurtzen duen bihurgailu bat.

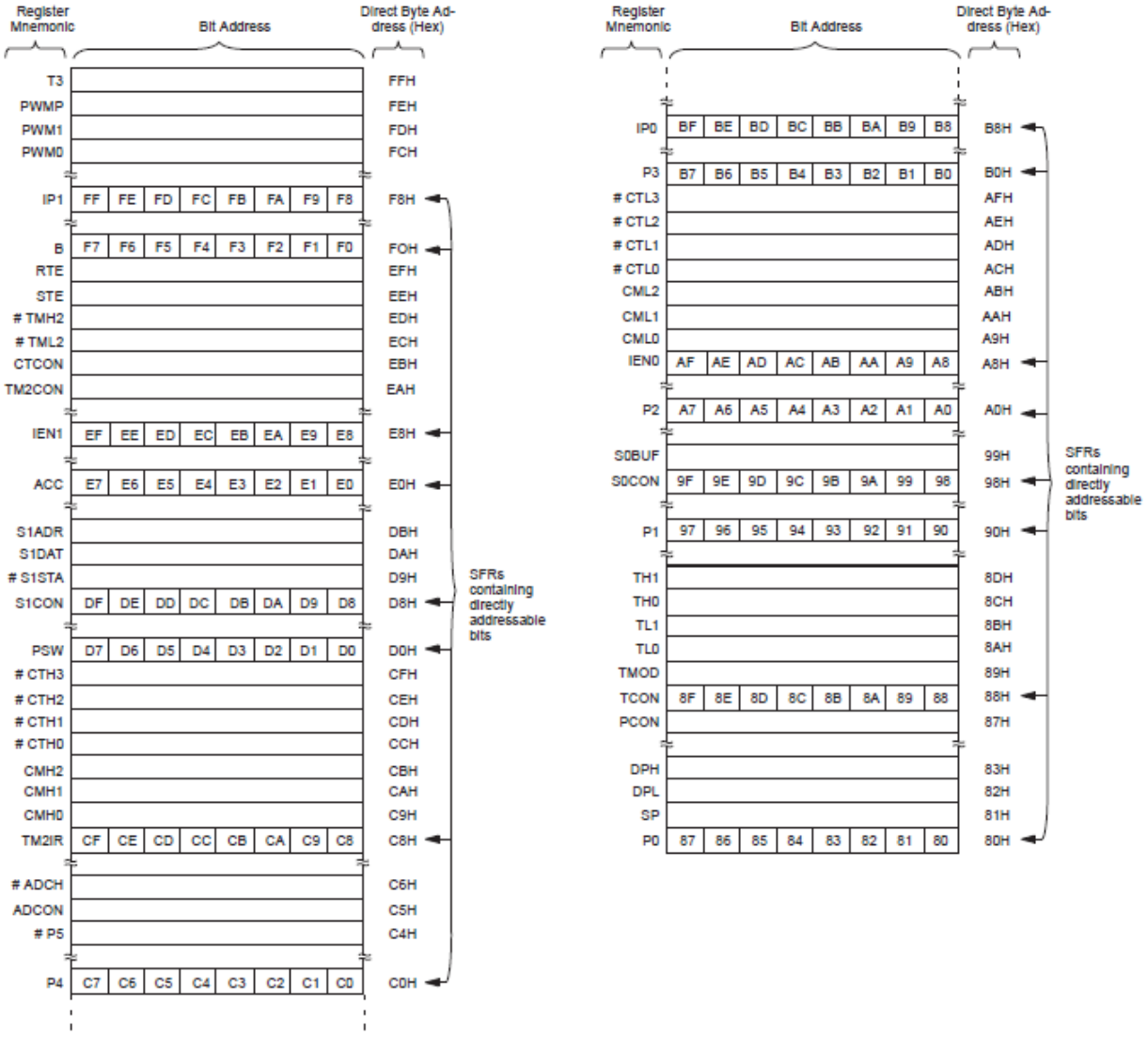
P5 portua erabili. Beraz, 8 ADC kanal ezberdin. Multipex.

Diskretua bai denboran, eta baita anplitudean ere.

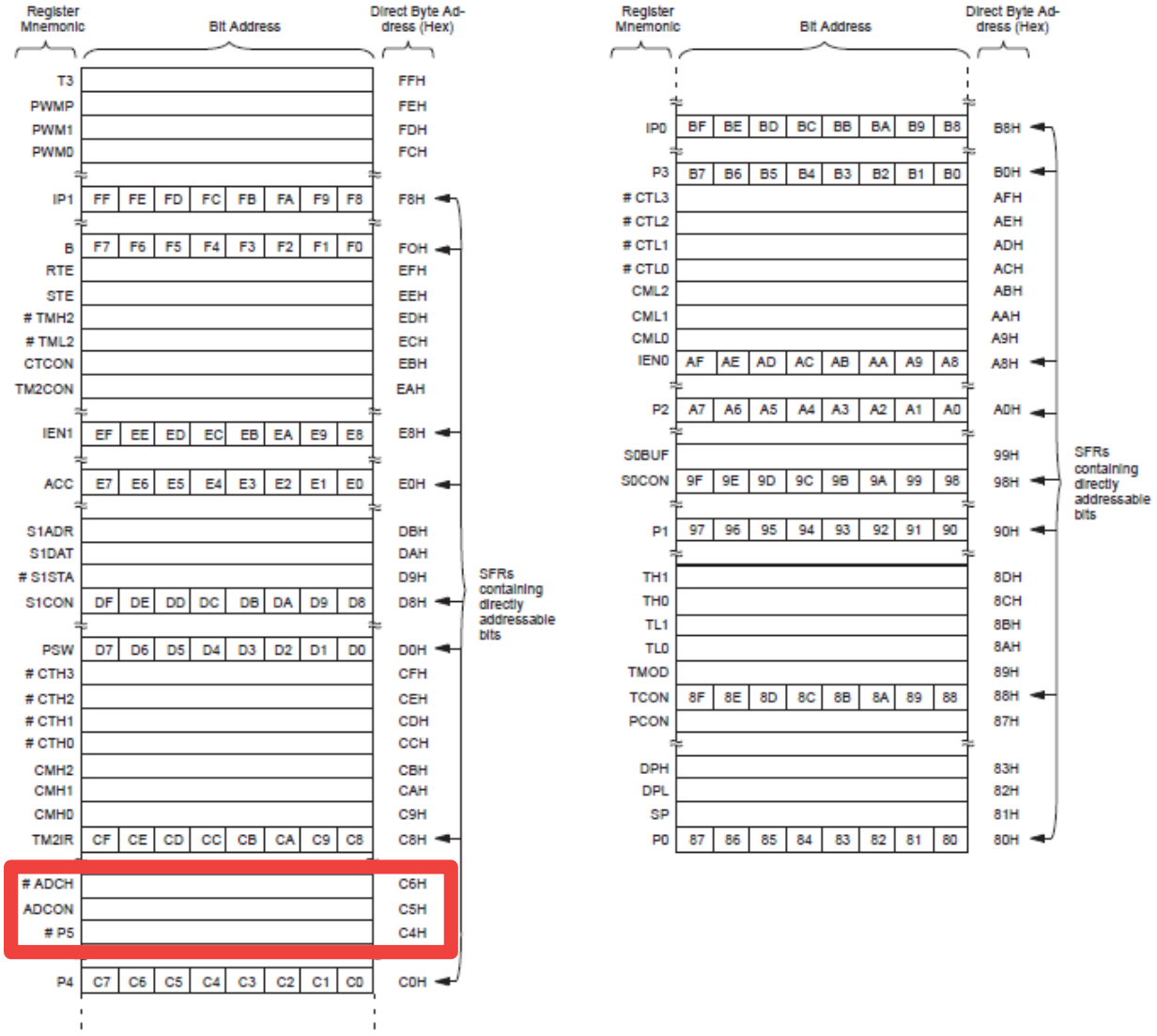
10 bit portu bakoitzeko. (2^(10))-eko resoluzioa. Tentsioan, Vcc/(2^(10))

80c552-ak 50 ziklo (fclk) behar prozesu osoa betetzeko.

ADC


Register Mnemonic
Bit Address
Direct Byte Address (Hex)
T3 FFH
PWMP FEH
PWM1 FDH
PWM0 FCH
IP1 FF FE FD FC FB FA F9 F8 F8H
B F7 F6 F5 F4 F3 F2 F1 F0 F0H
RTE EFH
STE EEH
TMH2 EDH
TML2 ECH
CTCON EBH
TM2CON EAH
IEN1 EF EE ED EC EB EA E9 E8 E8H
ACC E7 E6 E5 E4 E3 E2 E1 E0 E0H
S1ADR DBH
S1DAT DAH
S1STA D9H
S1CON DF DE DD DC DB DA D9 D8 D8H
PSW D7 D6 D5 D4 D3 D2 D1 D0 D0H
CTH3 CFH
CTH2 CEH
CTH1 CDH
CTH0 CCH
CMH2 CBH
CMH1 CAH
CMH0 C9H
TM2IR CF CE CD CC CB CA C9 C8 C8H
ADCH C6H
ADCON C5H
P5 C4H
P4 C7 C6 C5 C4 C3 C2 C1 C0 C0H
SFRs containing directly addressable bits
Register Mnemonic
Bit Address
Direct Byte Address (Hex)
IP0 BF BE BD BC BB BA B9 B8 B8H
P3 B7 B6 B5 B4 B3 B2 B1 B0 B0H
CTL3 AFH
CTL2 AEH
CTL1 ADH
CTL0 ACH
CML2 ABH
CML1 AAH
CML0 A9H
IEN0 AF AE AD AC AB AA A9 A8 A8H
P2 A7 A6 A5 A4 A3 A2 A1 A0 A0H
SDBUF 99H
SDCON 9F 9E 9D 9C 9B 9A 99 98 98H
P1 97 96 95 94 93 92 91 90 90H
TH1 8DH
TH0 8CH
TL1 8BH
TL0 8AH
TMOD 89H
TCON 8F 8E 8D 8C 8B 8A 89 88 88H
PCON 87H
DPH 83H
DPL 82H
SP 81H
P0 87 86 85 84 83 82 81 80 80H
SFRs containing directly addressable bits

ADC


Register Mnemonic
Bit Address
Direct Byte Address (Hex)
T3 FFH
PWMP FEH
PWM1 FDH
PWM0 FCH
IP1 FF FE FD FC FB FA F9 F8 F8H
B F7 F6 F5 F4 F3 F2 F1 F0 F0H
RTE EFH
STE EEH
TMH2 EDH
TML2 ECH
CTCON EBH
TM2CON EAH
IEN1 EF EE ED EC EB EA E9 E8 E8H
ACC E7 E6 E5 E4 E3 E2 E1 E0 E0H
S1ADR DBH
S1DAT DAH
S1STA D9H
S1CON DF DE DD DC DB DA D9 D8 D8H
PSW D7 D6 D5 D4 D3 D2 D1 D0 D0H
CTH3 CFH
CTH2 CEH
CTH1 CDH
CTH0 CCH
CMH2 CBH
CMH1 CAH
CMH0 C9H
TM2IR CF CE CD CC CB CA C9 C8 C8H
ADCH C6H
ADCON C5H
P5 C4H
P4 C7 C6 C5 C4 C3 C2 C1 C0 C0H
SFRs containing directly addressable bits
Register Mnemonic
Bit Address
Direct Byte Address (Hex)
IP0 BF BE BD BC BB BA B9 B8 B8H
P3 B7 B6 B5 B4 B3 B2 B1 B0 B0H
CTL3 AFH
CTL2 AEH
CTL1 ADH
CTL0 ACH
CML2 ABH
CML1 AAH
CML0 A9H
IEN0 AF AE AD AC AB AA A9 A8 A8H
P2 A7 A6 A5 A4 A3 A2 A1 A0 A0H
SDBUF 99H
SDCON 9F 9E 9D 9C 9B 9A 99 98 98H
P1 97 96 95 94 93 92 91 90 90H
TH1 8DH
TH0 8CH
TL1 8BH
TL0 8AH
TMOD 89H
TCON 8F 8E 8D 8C 8B 8A 89 88 88H
PCON 87H
DPH 83H
DPL 82H
SP 81H
P0 87 86 85 84 83 82 81 80 80H
SFRs containing directly addressable bits

ADC

Analog-to-Digital Conversion: Figure 35 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (Vin). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set and a value of "1" will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

During the next eight machine cycles, the voltage at the previously selected pin of port 5 is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000B). The output of the DAC (50% full scale) is compared to the input voltage Vin. If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000B or 01 0000 0000B, depending on the previous result), and VDAC is compared to Vin again. If the input voltage is greater than VDAC, then the bit being tested remains set; otherwise the bit being tested is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCON.7 (ADC.1) and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCI will be set and the ADCS status flag will be reset 50 (or 24) cycles after the command flip-flop (ADCS) is set.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (see Figure 37). An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

ADC

Analog-to-Digital Conversion: Figure 35 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (Vin). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set and a value of "1" will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

During the next eight machine cycles, the voltage at the previously selected pin of port 5 is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000B). The output of the DAC (50% full scale) is compared to the input voltage Vin. If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000B or 01 0000 0000B, depending on the previous result), and VDAC is compared to Vin again. If the input voltage is greater than VDAC, then the bit being tested remains set; otherwise the bit being tested is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCON.7 (ADC.1) and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCI will be set and the ADCS status flag will be reset 50 (or 24) cycles after the command flip-flop (ADCS) is set.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (see Figure 37). An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

ADC

ADCON (C5H)	7	6	5	4	3	2	1	0
	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0
	(MSB)							(LSB)

Bit	Symbol	Function
ADCON.7	ADC.1	Bit 1 of ADC result
ADCON.6	ADC.0	Bit 0 of ADC result
ADCON.5	ADEX	Enable external start of conversion by STADC 0 = Conversion can be started by software only (by setting ADCS) 1 = Conversion can be started by software or externally (by a rising edge on STADC)
ADCON.4	ADCI	ADC interrupt flag: this flag is set when an A/D conversion result is ready to be read. An interrupt is invoked if it is enabled. The flag may be cleared by the interrupt service routine. While this flag is set, the ADC cannot start a new conversion. ADCI cannot be set by software.
ADCON.3	ADCS	ADC start and status: setting this bit starts an A/D conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset immediately after the interrupt flag has been set. ADCS cannot be reset by software. A new conversion may not be started while either ADCS or ADCI is high.

ADCI	ADCS	ADC Status
0	0	ADC not busy; a conversion can be started
0	1	ADC busy; start of a new conversion is blocked
1	0	Conversion completed; start of a new conversion requires ADCI=0
1	1	Conversion completed; start of a new conversion requires ADCI=0

If ADCI is cleared by software while ADCS is set at the same time, a new A/D conversion with the same channel number may be started. But it is recommended to reset ADCI *before* ADCS is set.

ADCON.2 ADCON.1 ADCON.0	AADR2 AADR1 AADR0	Analogue input select: this binary coded address selects one of the eight analogue port bits of P5 to be input to the converter. It can only be changed when ADCI and ADCS are both LOW.

AADR2	AADR1	AADR0	Selected Analog Channel
0	0	0	ADC0 (P5.0)
0	0	1	ADC1 (P5.1)
0	1	0	ADC2 (P5.2)
0	1	1	ADC3 (P5.3)
1	0	0	ADC4 (P5.4)
1	0	1	ADC5 (P5.5)
1	1	0	ADC6 (P5.6)
1	1	1	ADC7 (P5.7)

Figure 37. ADC Control Register (ADCON)

ADC

	7	6	5	4	3	2	1	0
ADCON (C5H)	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0
	(MSB)							(LSB)

Bit	Symbol	Function
ADCON.7	ADC.1	Bit 1 of ADC result
ADCON.5	ADEX	Enable external start of conversion by STADC 0 = Conversion can be started by software only (by setting ADCS) 1 = Conversion can be started by software or externally (by a rising edge on STADC)
ADCON.4	ADCI	ADC interrupt flag: this flag is set when an A/D conversion result is ready to be read. An interrupt is invoked if it is enabled. The flag may be cleared by the interrupt service routine. While this flag is set, the ADC cannot start a new conversion. ADCI cannot be set by software.
ADCON.3	ADCS	ADC start and status: setting this bit starts an A/D conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset immediately after the interrupt flag has been set. ADCS cannot be reset by software. A new conversion may not be started while either ADCS or ADCI is high.

ADCI	ADCS	ADC Status
0	0	ADC not busy; a conversion can be started
0	1	ADC busy; start of a new conversion is blocked
1	0	Conversion completed; start of a new conversion requires ADCI=0
1	1	Conversion completed; start of a new conversion requires ADCI=0

If ADCI is cleared by software while ADCS is set at the same time, a new A/D conversion with the same channel number may be started. But it is recommended to reset ADCI *before* ADCS is set.

ADCON.2 ADCON.1 ADCON.0	AADR2 AADR1 AADR0	Analogue input select: this binary coded address selects one of the eight analogue port bits of P5 to be input to the converter. It can only be changed when ADCI and ADCS are both LOW.

AADR2	AADR1	AADR0	Selected Analog Channel
0	0	0	ADC0 (P5.0)
0	0	1	ADC1 (P5.1)
0	1	0	ADC2 (P5.2)
0	1	1	ADC3 (P5.3)
1	0	0	ADC4 (P5.4)
1	0	1	ADC5 (P5.5)
1	1	0	ADC6 (P5.6)
1	1	1	ADC7 (P5.7)

Figure 37. ADC Control Register (ADCON)

ADC

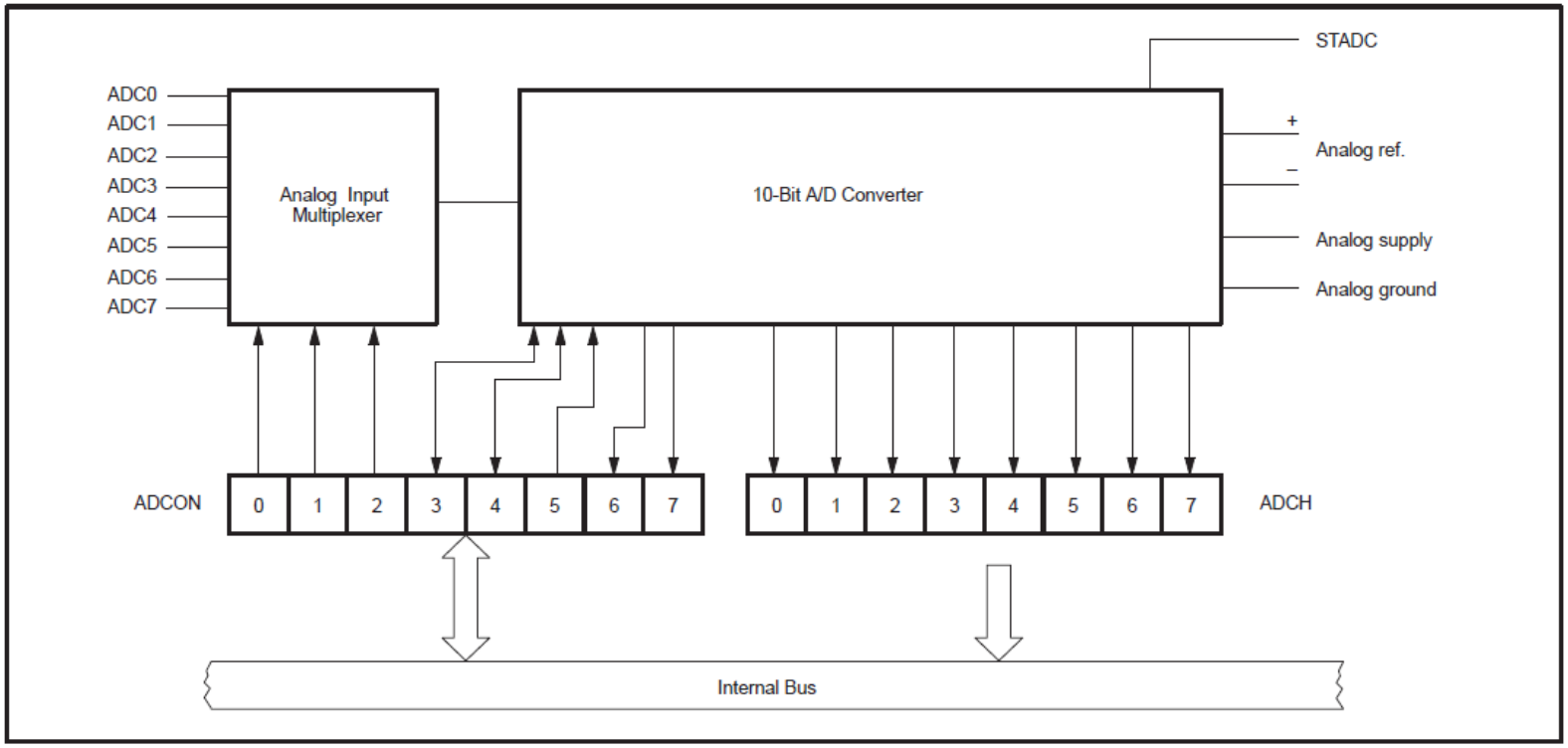


Figure 34. Functional Diagram of Analog Input Circuitry

ADC

Analog-to-Digital Conversion: Figure 35 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (Vin). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set and a value of "1" will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

During the next eight machine cycles, the voltage at the previously selected pin of port 5 is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000B). The output of the DAC (50% full scale) is compared to the input voltage Vin. If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000B or 01 0000 0000B, depending on the previous result), and VDAC is compared to Vin again. If the input voltage is greater than VDAC, then the bit being tested remains set; otherwise the bit being tested is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCON.7 (ADC.1) and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCI will be set and the ADCS status flag will be reset 50 (or 24) cycles after the command flip-flop (ADCS) is set.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (see Figure 37). An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

ADC

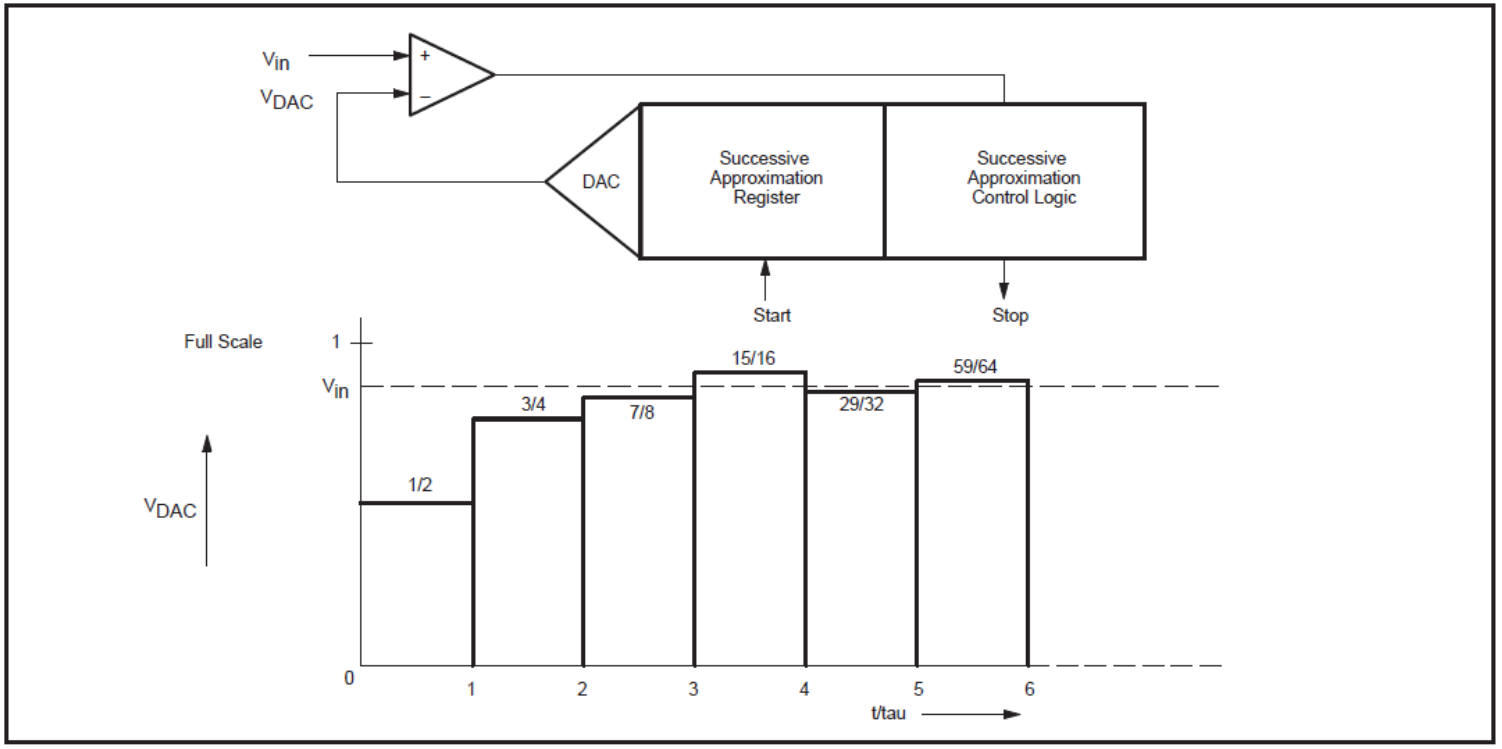
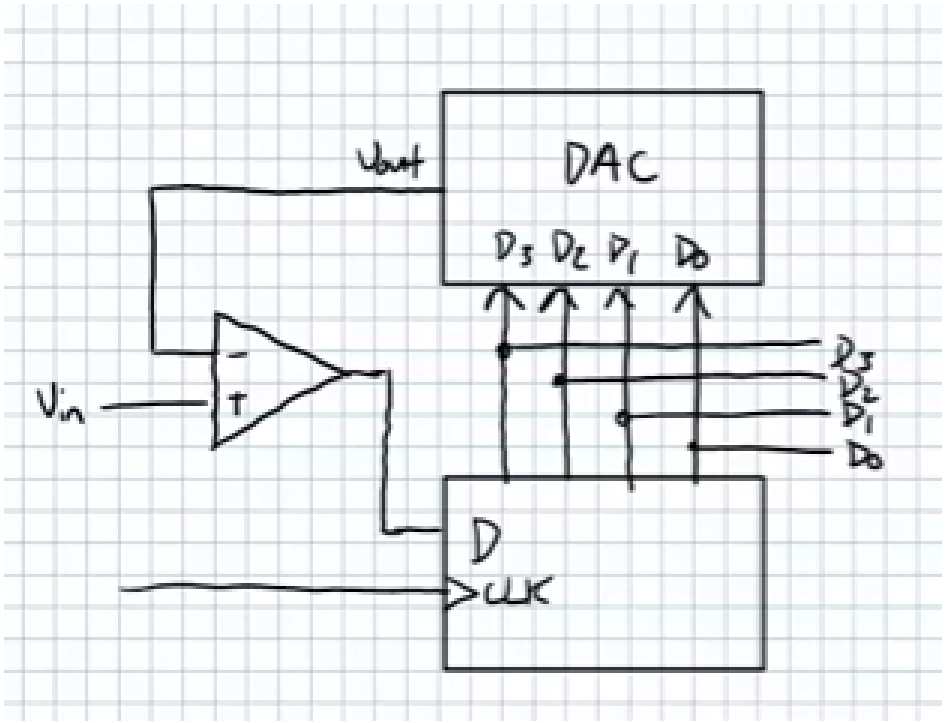


Figure 35. Successive Approximation ADC

ADC

Successive Approximation ADC: MSB-tik, LSB-ra bitez bit erreferentzi balioarekin konparatu.



ADC

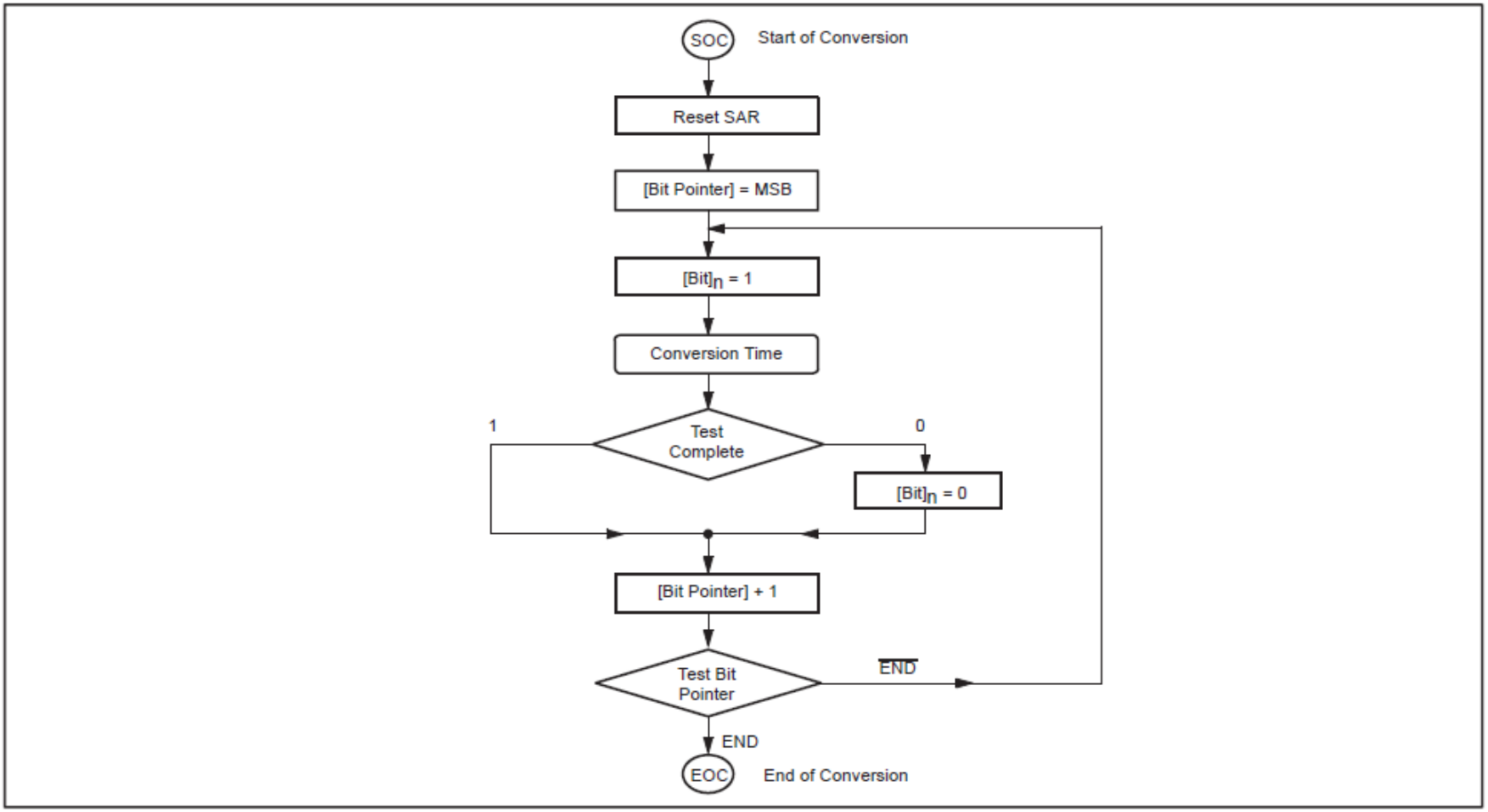


Figure 36. A/D Conversion Flowchart

ADC

Zein resoluzio eta eskala limite eukiko genituzke 10 bit beharrean 8 bit erabiliko bagenitu?

Ze bit erabili beharko lirateke bit kopuru txikiagoa erabili nahi izanez gero?

ADC

Beraz, zelan kalkulatzen da resoluzioa?

ADC

Beraz, zelan kalkulatzen da resoluzioa?

Resoluzioa: Vref / (2^(Nbits))

ADC

Beraz, zelan kalkulatzen da resoluzioa?

Resoluzioa: Vref / (2^(Nbits))

Eta zelan kalkulatzen da neurtutako balioa (SAR ADC)?

ADC

Eta zelan kalkulatzen da neurtutako balioa (SAR ADC)?

ADC Resolution and Analog Supply: Figure 38 shows how the ADC is realized. The ADC has its own supply pins (AV_{DD} and AV_{SS}) and two pins (Vref+ and Vref–) connected to each end of the DAC's resistance-ladder. The ladder has 1023 equally spaced taps, separated by a resistance of "R". The first tap is located 0.5 x R above Vref–, and the last tap is located 1.5 x R below Vref+. This gives a total ladder resistance of 1024 x R. This structure ensures that the DAC is monotonic and results in a symmetrical quantization error as shown in Figure 40.

For input voltages between Vref– and (Vref–) + 1/2 LSB, the 10-bit result of an A/D conversion will be 00 0000 0000B = 000H. For input voltages between (Vref+) – 3/2 LSB and Vref+, the result of a conversion will be 11 1111 1111B = 3FFH. AVref+ and AVref– may be between AV_{DD} + 0.2V and AV_{SS} – 0.2V. AVref+ should be positive with respect to AVref–, and the input voltage (Vin) should be between AVref+ and AVref–. If the analog input voltage range is from 2V to 4V, then 10-bit resolution can be obtained over this range if AVref+ = 4V and AVref– = 2V.

The result can always be calculated from the following formula:

$$\text{Result} = 1024 \times \frac{V_{IN} - AV_{ref-}}{AV_{ref+} - AV_{ref-}}$$

ADC

Eta zelan kalkulatzen da neurtutako balioa (SAR ADC)?

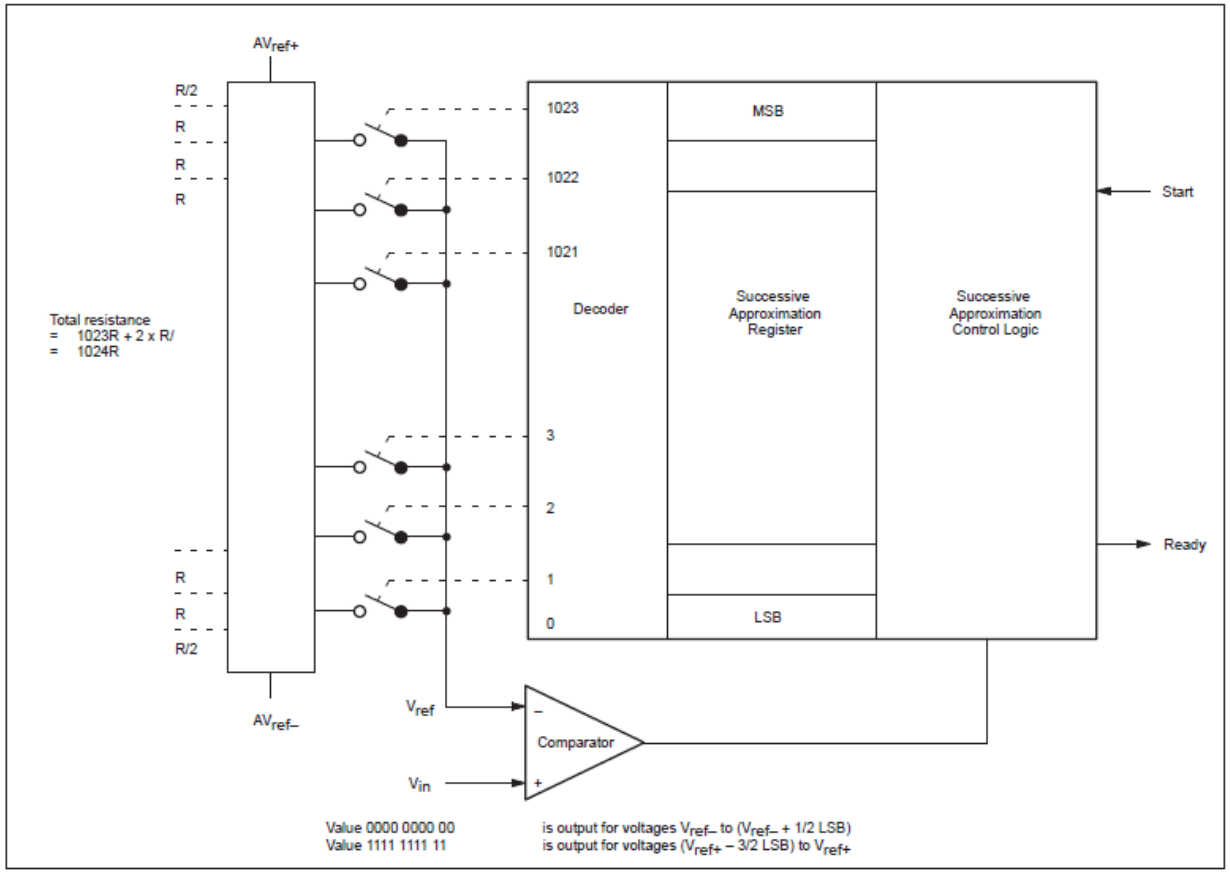


Figure 38. ADC Realization

ADC

Errorea positibo eta negatiboa lortu horrela!

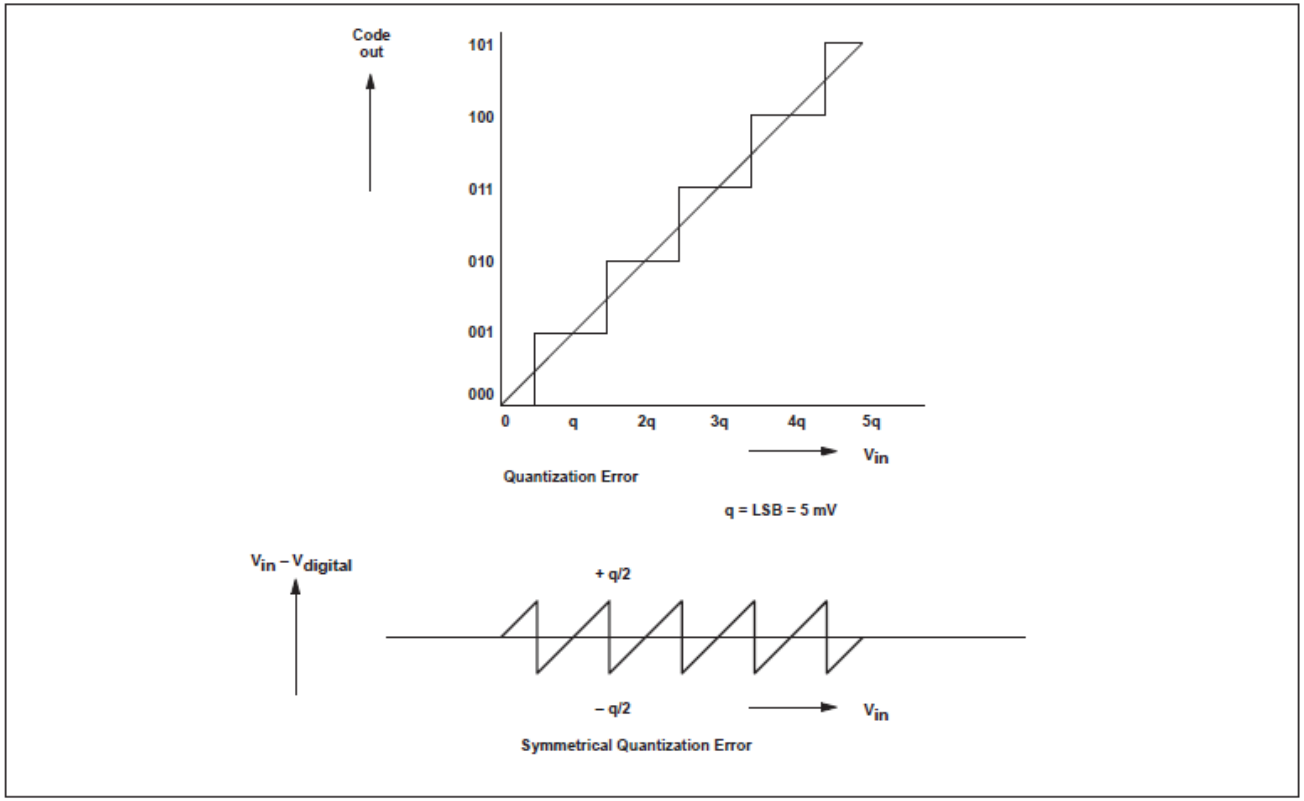


Figure 40. Effective Conversion Characteristic

ADC

80c552-a 5V-tara elikatzen baldin badugu:

Zelan konfiguratu behar da ADC-a (ADCON), P5.6-an, 0V-tik 2.5V-ra doan seinale bat neurtu nahi bada? 10 bitetatik zenbat bit erabili behar dira 5V-tara elikatuta badago?

Ze eskala resoluzioa eukiko genuke (Vcc = 5V), ADC-an 10 bit badauzkagu? Eta bakarrik 8 bit nahi izanez gero eta elikadura 5V-tan baldin badago?

ADC

Beraz, zelan kalkulatzen da resoluzioa?

Resol = Vref / (2^(Nbits))

Eta zelan kalkulatzen da neurtutako balioa (SAR ADC)?

MeasADC = Floor (Vmeasure / resol) * resol

ADC

Beraz, zelan kalkulatzen da resoluzioa?

Resol = Vref / (2^(Nbits))

Eta zelan kalkulatzen da neurtutako balioa (SAR ADC)?

MeasADC = Floor (Vmeasure / resol) * resol

Meas80C552 = Floor ((Vmeasure + resol/2) / resol) * resol

ADC

80c552-a Vref-tara elikatzen baldin badugu, zein da resoluzioa eta zenbat neurtuko genuke ADC bit kopuru diferenteekin:

Vref	Vmeas.	ADC 2 bit	ADC 4 bit	ADC 8 bit	ADC 10 bit	80c552
5V	3.3V					
5V	4.88V					
5V	2.5V					
5V	0.004V					
3.3V	4.88V					
3.3V	2.5V					
5V	4.99V					
5V	1.259V					

ADC

Interrupt priority?

IP0 (B8H)	7	6	5	4	3	2	1	0
	–	PAD	PS1	PS0	PT1	PX1	PT0	PX0
	(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
IP0.7	–	Unused
IP0.6	PAD	ADC interrupt priority level
IP0.5	PS1	SIO1 (I^2C) interrupt priority level
IP0.4	PS0	SIO0 (UART) interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

SU00763

Figure 30. Interrupt Priority Register (IP0)

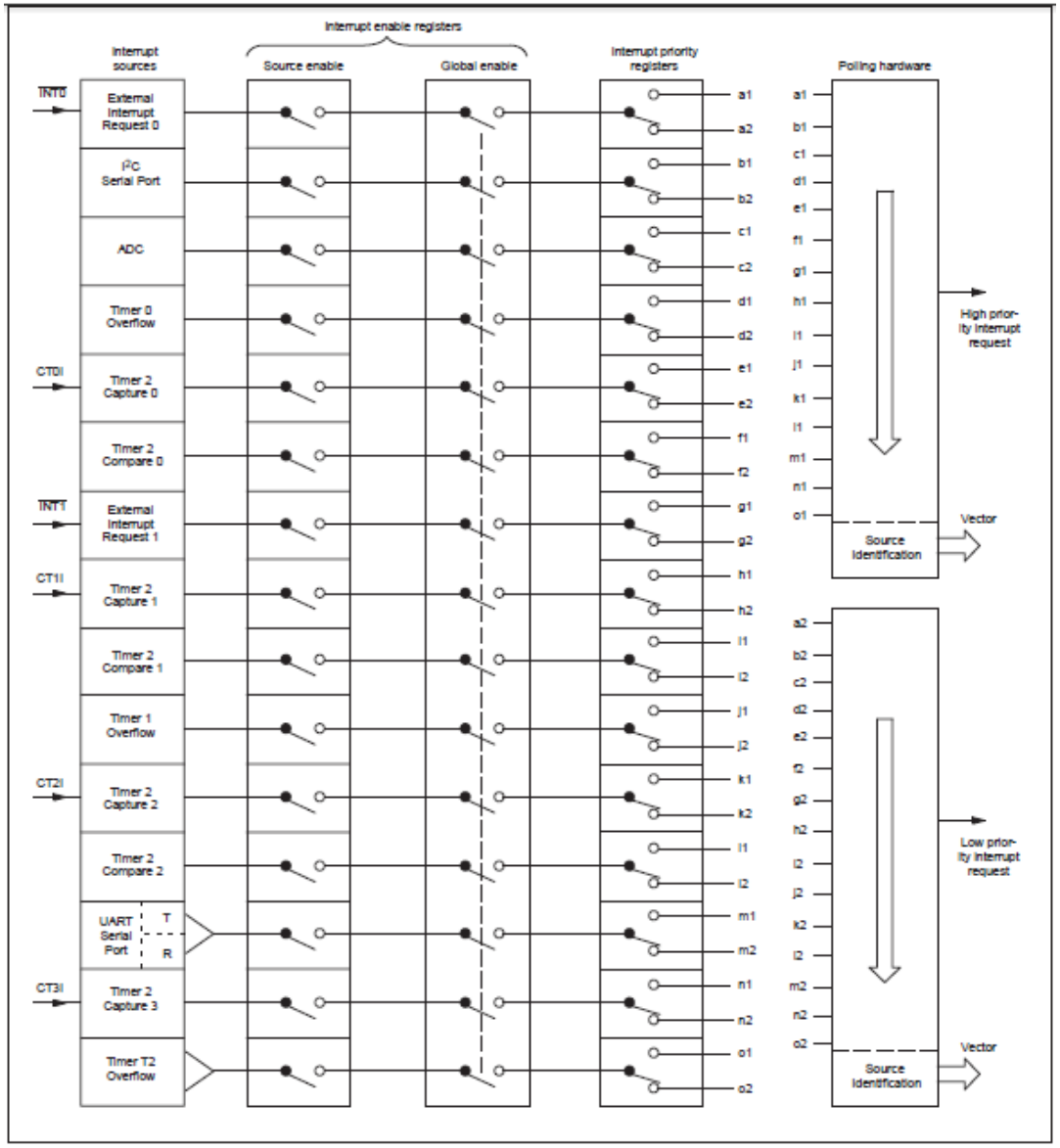
IP1 (F8H)	7	6	5	4	3	2	1	0
	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0
	(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
IP1.7	PT2	T2 overflow interrupt(s) priority level
IP1.6	PCM2	T2 comparator 2 interrupt priority level
IP1.5	PCM1	T2 comparator 1 interrupt priority level
IP1.4	PCM0	T2 comparator 0 interrupt priority level
IP1.3	PCT3	T2 capture register 3 interrupt priority level
IP1.2	PCT2	T2 capture register 2 interrupt priority level
IP1.1	PCT1	T2 capture register 1 interrupt priority level
IP1.0	PCT0	T2 capture register 0 interrupt priority level

SU00764

Figure 31. Interrupt Priority Register (IP1)

ADC


Interrupt enable registers
Interrupt sources
Source enable
Global enable
Interrupt priority registers
Polling hardware
INT0
External Interrupt Request 0
I²C Serial Port
ADC
Timer 0 Overflow
CT0I
Timer 2 Capture 0
Timer 2 Compare 0
INT1
External Interrupt Request 1
CT1I
Timer 2 Capture 1
Timer 2 Compare 1
Timer 1 Overflow
CT2I
Timer 2 Capture 2
Timer 2 Compare 2
UART Serial Port
T
R
CT3I
Timer 2 Capture 3
Timer T2 Overflow
a1 a2 b1 b2 c1 c2 d1 d2 e1 e2 f1 f2 g1 g2 h1 h2 i1 i2 j1 j2 k1 k2 l1 l2 m1 m2 n1 n2 o1 o2
High priority interrupt request
Vector
Source Identification
Low priority interrupt request
Vector
Source Identification

HELBURUAK

Aldagai analogikoekin zelan lan egin daitekeen ulertzea

80c552-aren ADC-a zelan konfiguratzen den ikastea

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Teknologia Elektronikoa Saila

KONPUTAGAILUEN ARKITEKTURA
80c552 - ADC

Kudeaketa eta Informazio Sistemen Informatikaren Ingenieritzako Graduaren 3. maila

Irakaslea: Alain Sanchez
(alain.sanchez@ehu.eus)

2019-2020, 1. lauhilabetea