

# Analogové vstupy a výstupy

**Mikroprocesorové a vestavěné systémy (IMP)**

Přednáší: doc. Ing. Richard Růžička, Ph. D.

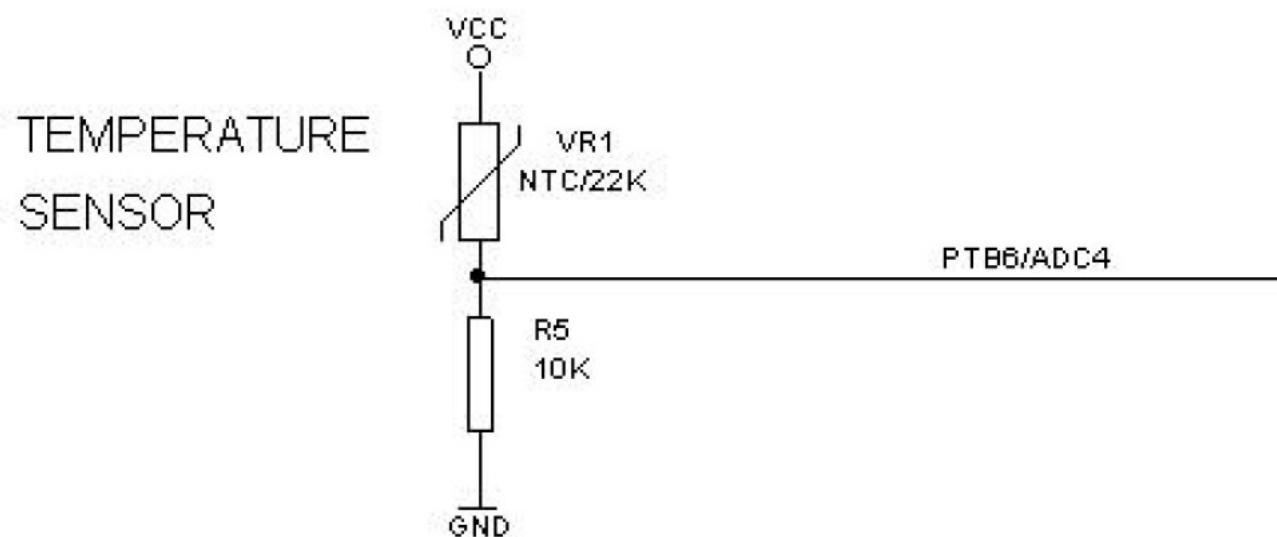
Fakulta informačních technologií VUT v Brně

# Analogový vstup

- Signály ze spojitých snímačů
  - teplota
  - tlak
  - osvětlení
  - odpor (např. odporový touchscreen)
  - napětí/proud
- Zpětná vazba
- „Rozšíření“ počtu číslicových vstupů

# Analogový vstup - příklad

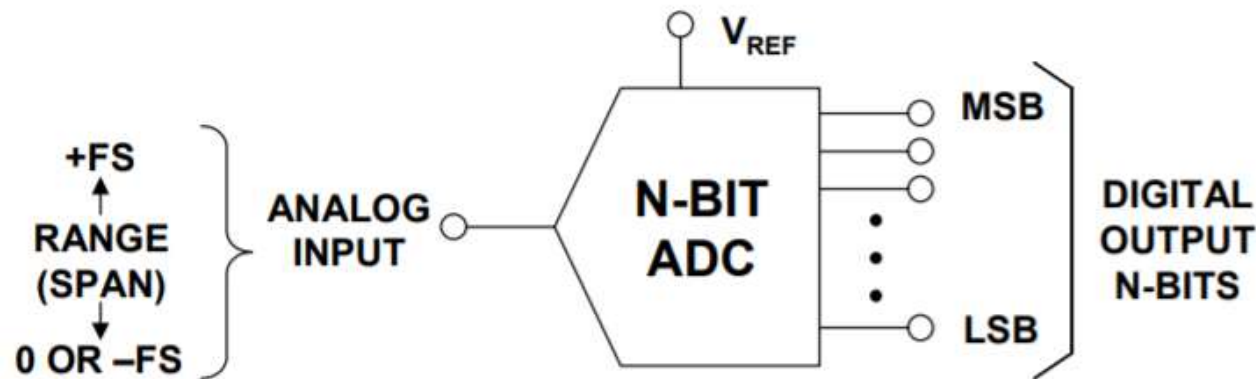
- Měření teploty



# Analogový vstup MCU

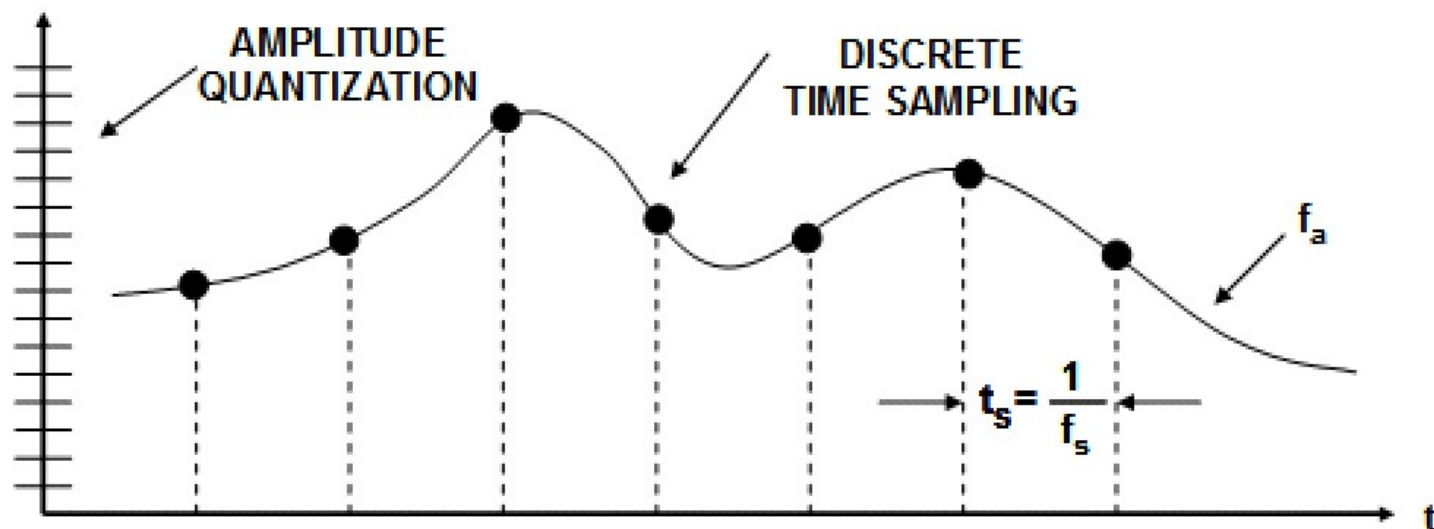
- MCU je číslicový, **analogovou hodnotu je třeba vyjádřit binárním číslem**, aby s ní bylo možno dále pracovat.
- Potřebujeme **A/D převodník**.
- A/D převodník (Analog to Digital Converter, ADC) je typickým modulem na čipu dnešních MCU.
- Někdy stačí jen vědět, **zda analogová hodnota je větší či menší než nějaká mez** – k vyjádření stačí dvoustavová hodnota, kterou lze získat **analogovým komparátorem**.

# Analogově-číslicový převodník – Analogue to Digital Converter, ADC



FS = Full Scale

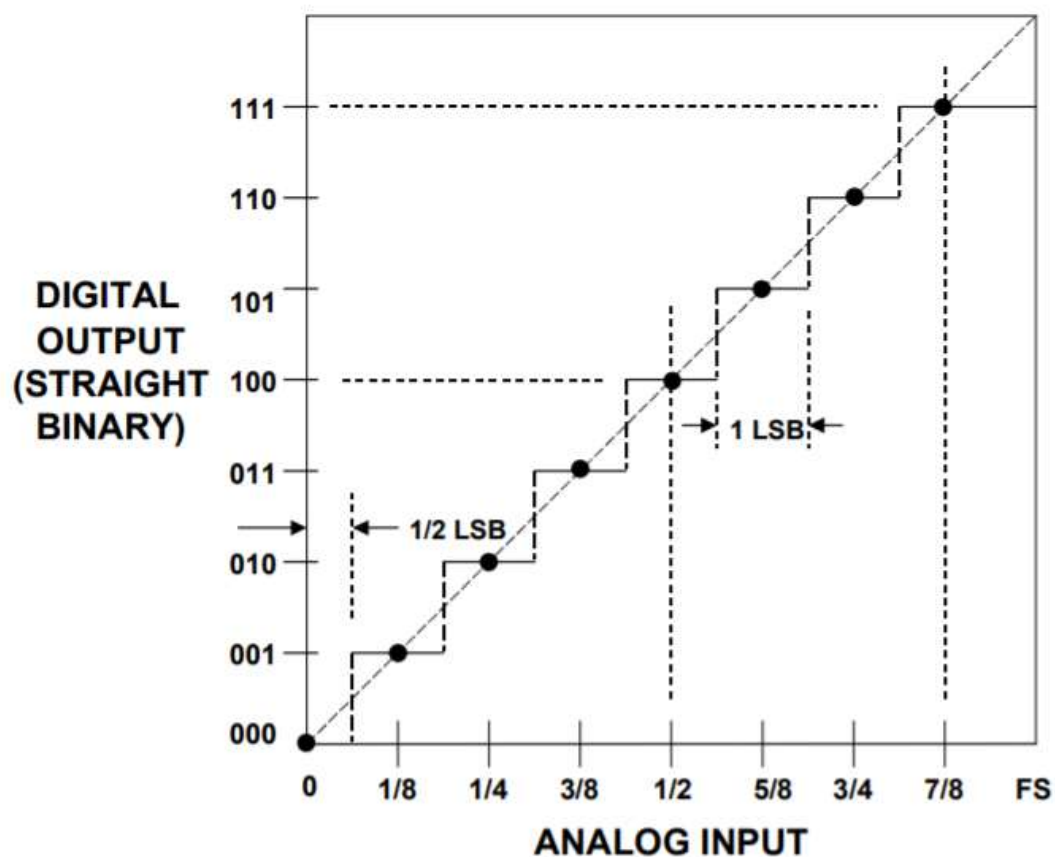
# Diskretizace v napěťové i časové oblasti



Důsledky:

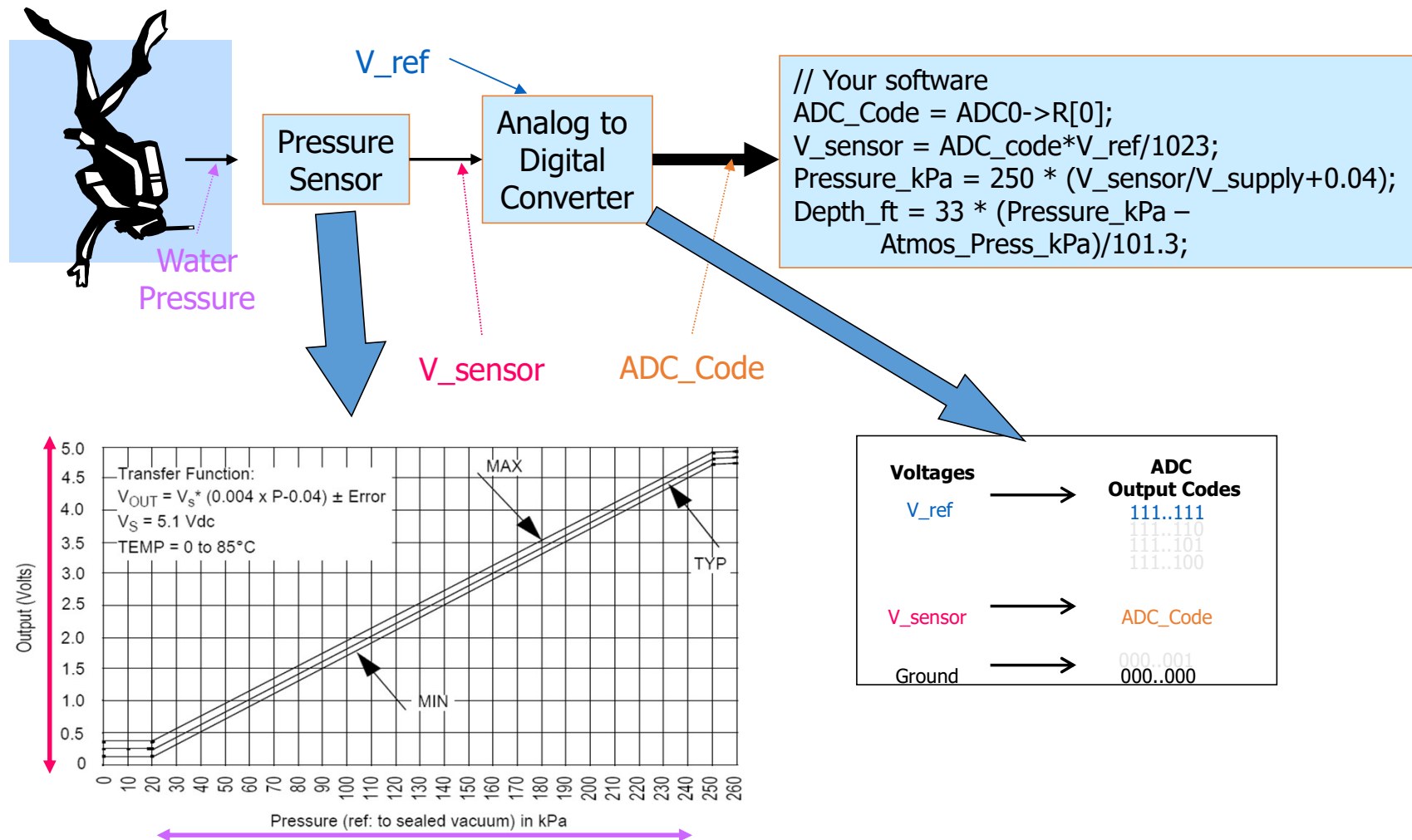
- teoreticky ztráta informace:
- diskretizace v napěťové oblasti přináší tzv. kvantizační šum
- diskretizace v časové oblasti může způsobit třeba aliasing

# Přenosová funkce ADC (pro jednoduchost tříbitového)



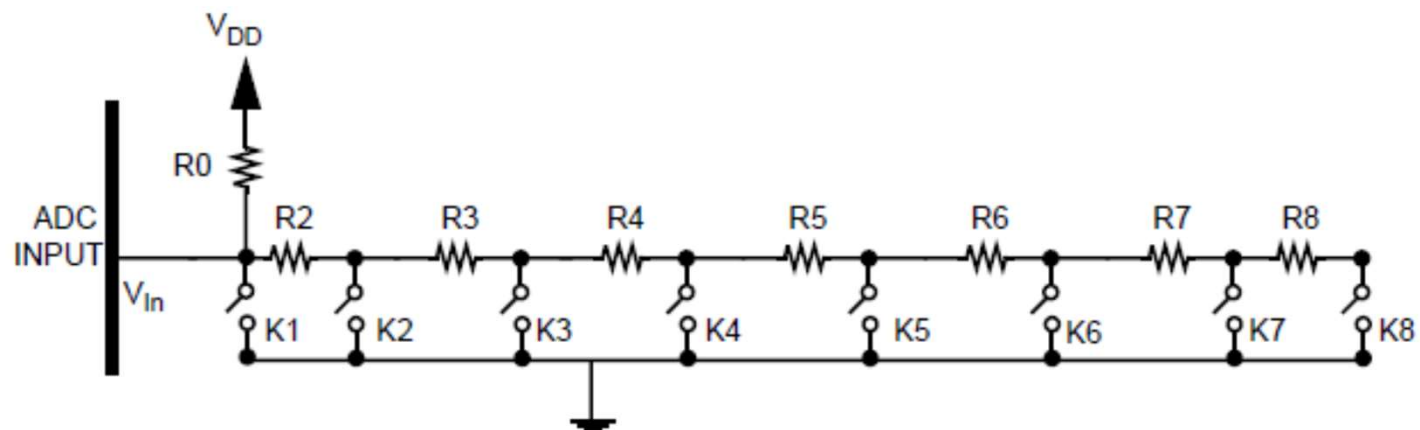
**Figure 2.5:** Transfer Function for Ideal Unipolar 3-bit ADC

# Příklad kompletního řetězce snímání analogové veličiny

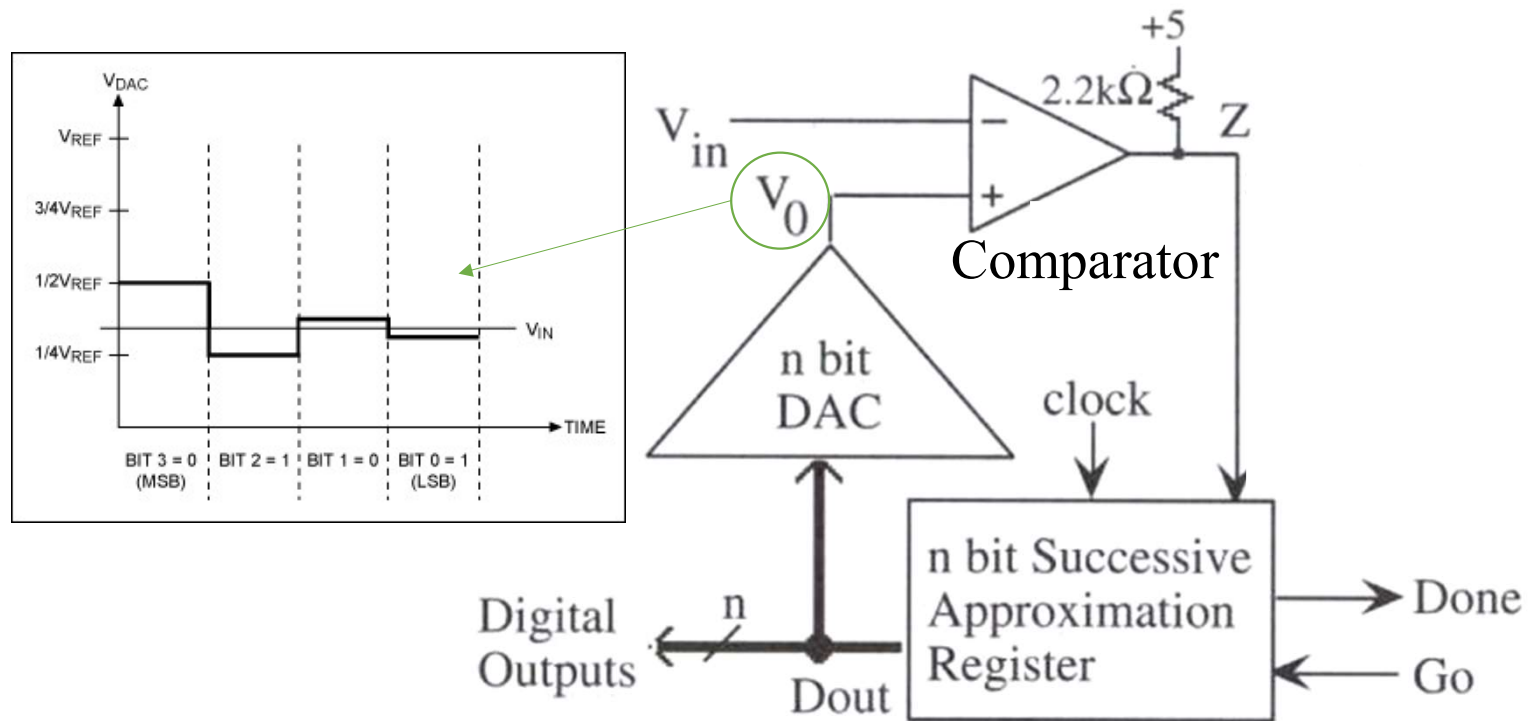




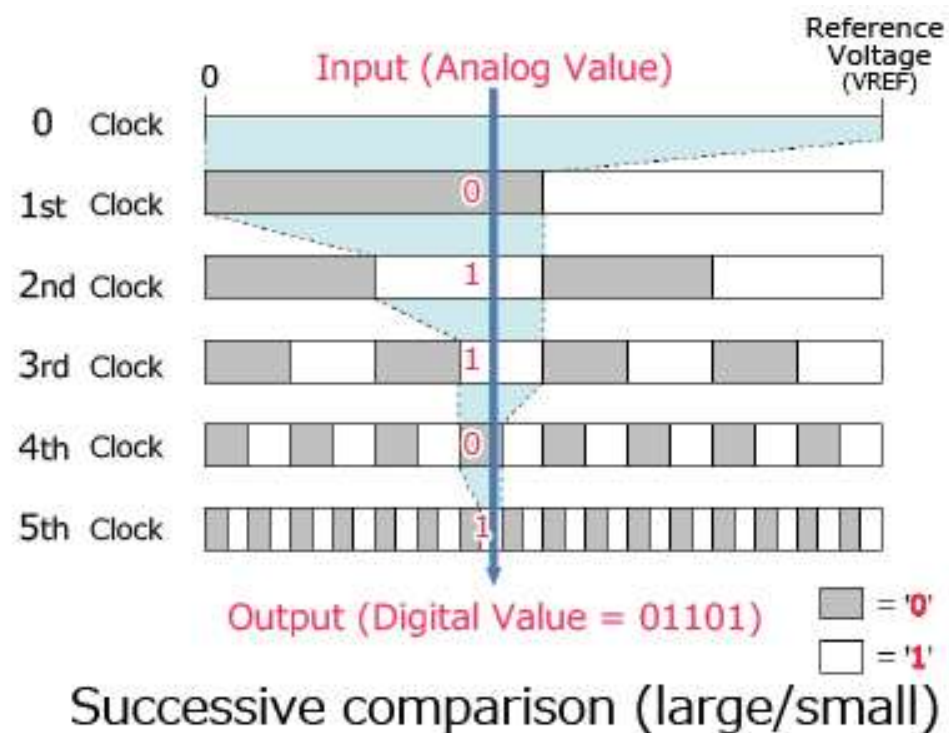
# Rozšíření digitálních vstupů



# Princip aproximačního ADC



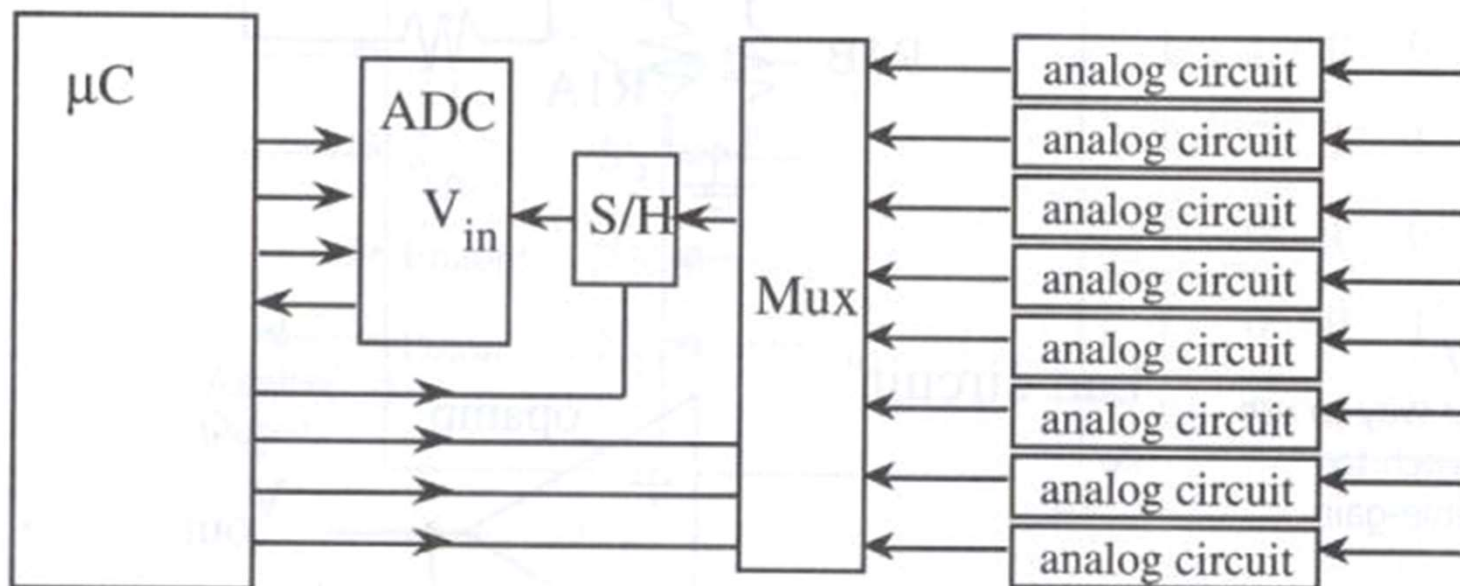
# Princip aproximačního ADC



# Je třeba si uvědomit:

- Převod analogové hodnoty na binární číslo pomocí aproximačního ADC je proces, který trvá nějaký čas – pro  $n$ -bitové číslo je třeba (nejméně)  $n$  kroků.
- Převod je taktován hodinami – jejich frekvence určuje, jak rychle dostáváme vzorky vstupního signálu (případně jak často, pokud vzorkujeme „co to dá“).
- Pokud se vstupní analogová hodnota změní dříve, než je převod dokončen, výsledek je nejspíš špatně (řeší se vzorkováním před převodem).
- Přesnost převodu je dále závislá na kvalitě (linearitě) D/A převodníku – přesný D/A převodník je velmi drahá komponenta.

## Více kanálů ADC

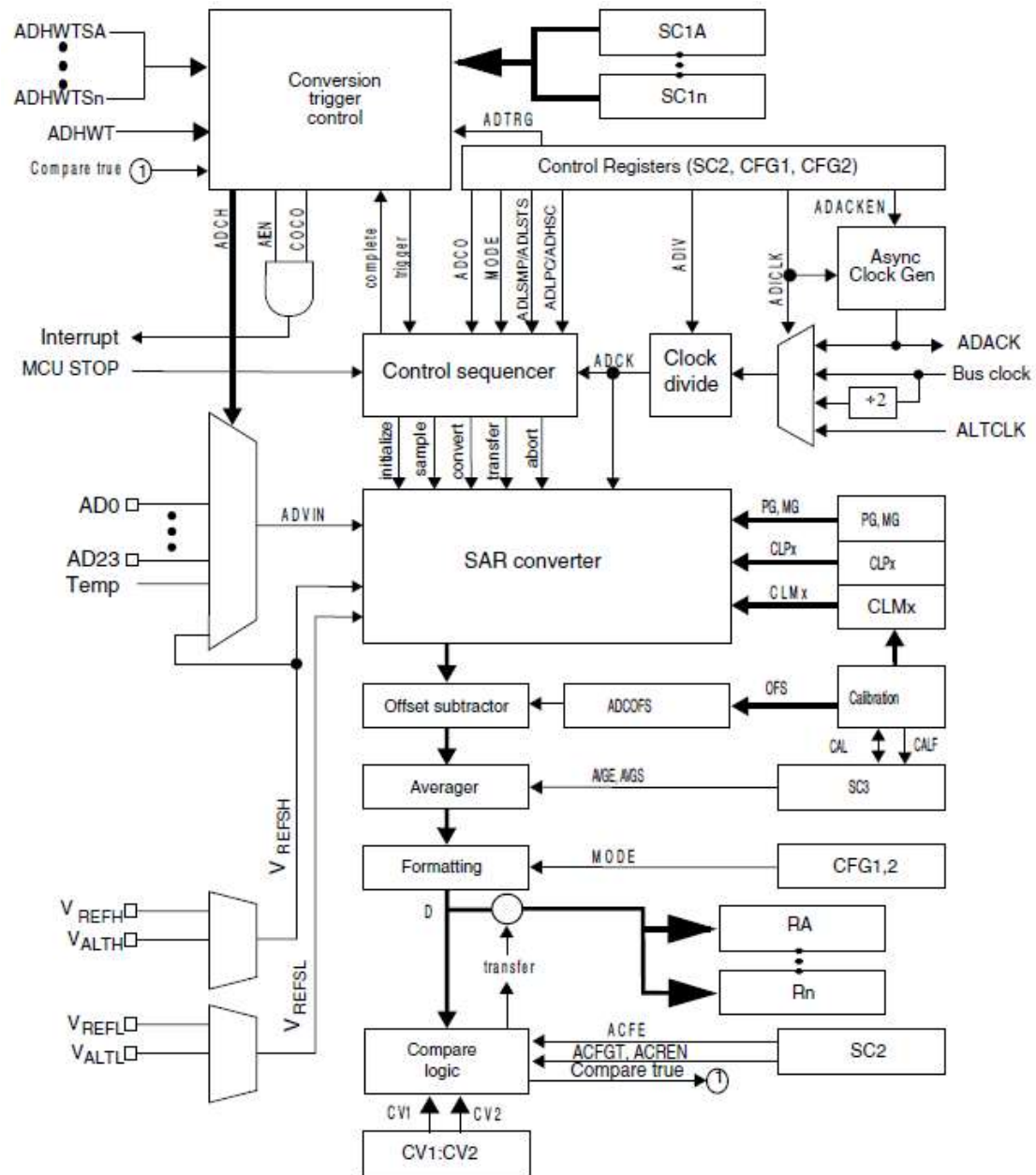


Je typické, že se používá jeden ADC pro více snímaných veličin – vstup se přepíná multiplexorem.

# Analogově-číslicový převodník

Modul ADC

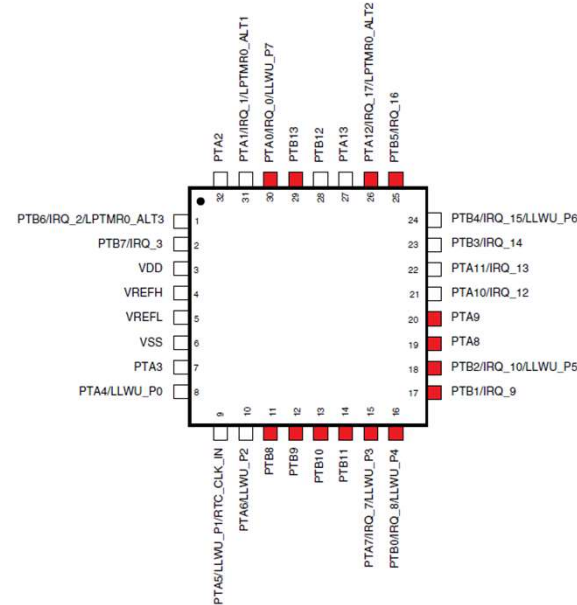
# Struktura ADC modulu MCU Kinetis



# Vstupy ADC

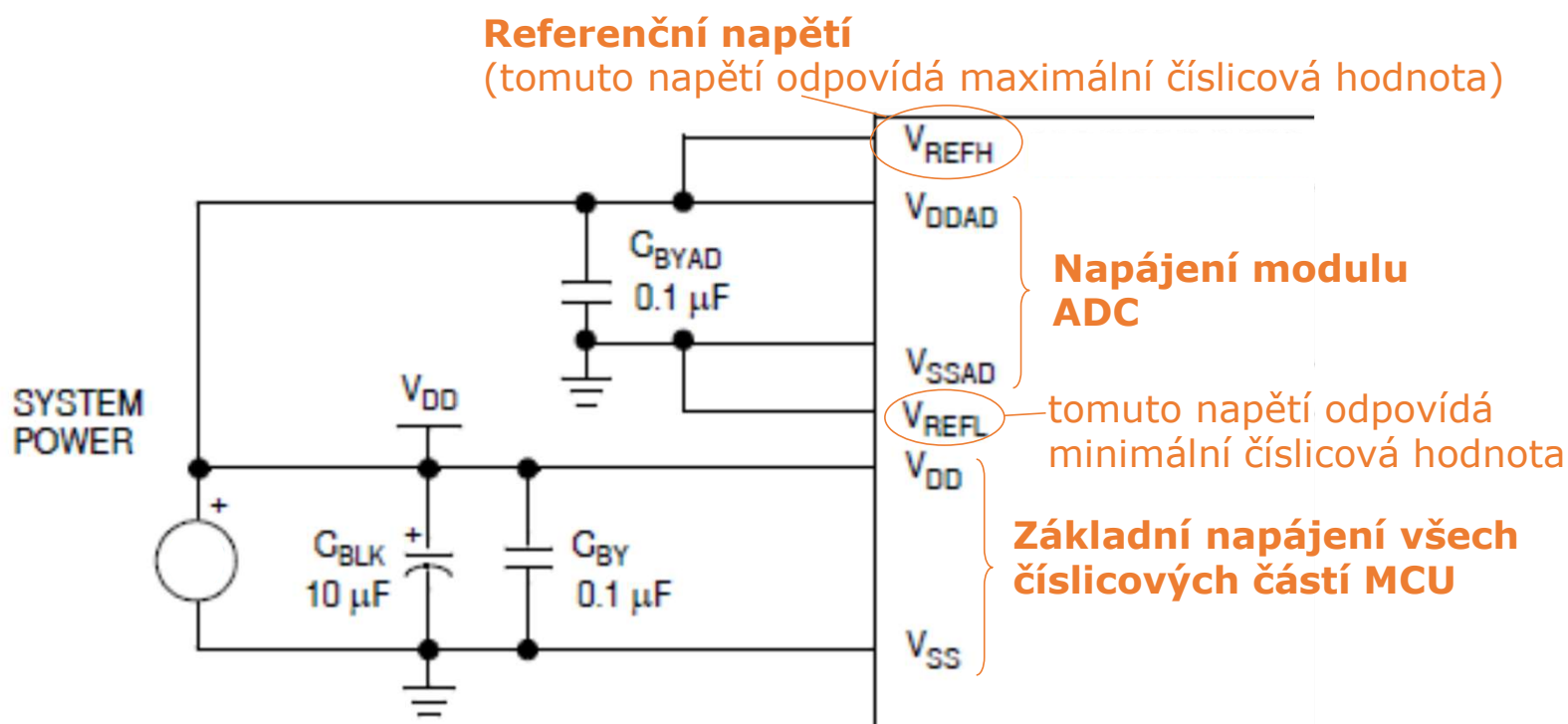
- Teoreticky může být až 24 (AD0 – AD23), prakticky je to omezeno použitým pouzdrem.
- Některé vstupy jsou zapojeny „uvnitř“ (referenční zdroj, teplotní senzor).

Příklad:  
KL05Z32VLC4  
(MCU na kitu v laboratoři)





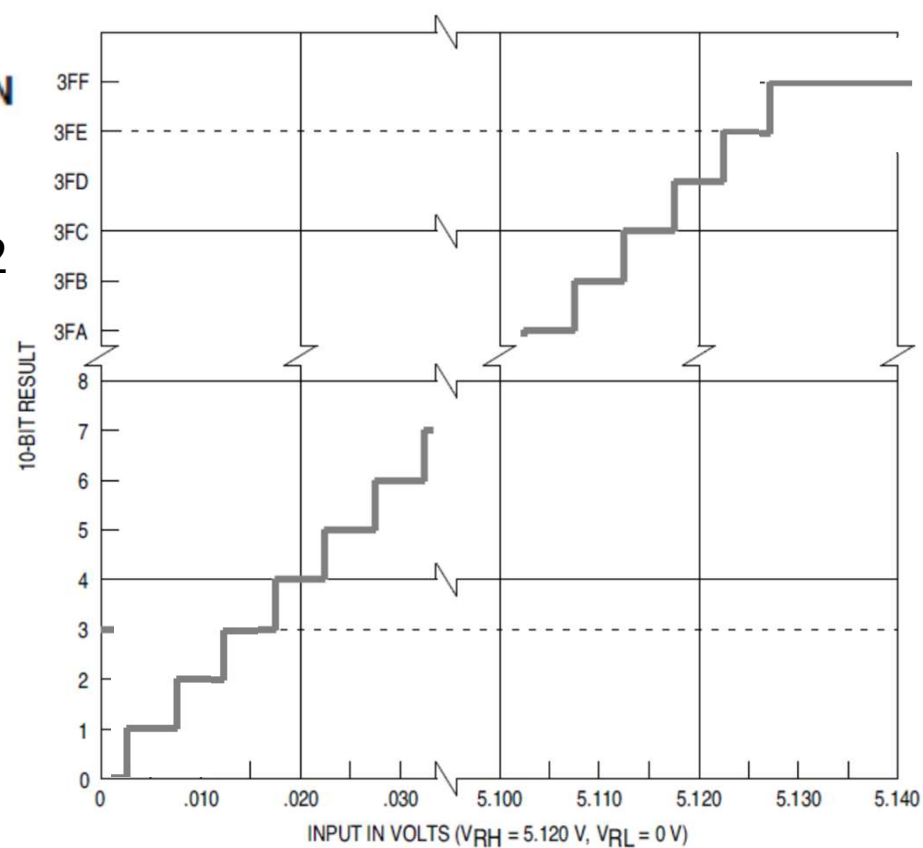
# Napájení ADC modulu



# Rozlišení převodníku

$$1 \text{ lsb} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N$$

N = 8 nebo 10 nebo 12



RESOLUTION N	$2^N$	VOLTAGE (10V FS)	ppm FS	% FS	dB FS
2-bit	4	2.5 V	250,000	25	- 12
4-bit	16	625 mV	62,500	6.25	- 24
6-bit	64	156 mV	15,625	1.56	- 36
8-bit	256	39.1 mV	3,906	0.39	- 48
10-bit	1,024	9.77 mV (10 mV)	977	0.098	- 60
12-bit	4,096	2.44 mV	244	0.024	- 72
14-bit	16,384	610 $\mu$ V	61	0.0061	- 84
16-bit	65,536	153 $\mu$ V	15	0.0015	- 96
18-bit	262,144	38 $\mu$ V	4	0.0004	- 108
20-bit	1,048,576	9.54 $\mu$ V (10 $\mu$ V)	1	0.0001	- 120
22-bit	4,194,304	2.38 $\mu$ V	0.24	0.000024	- 132
24-bit	16,777,216	596 nV*	0.06	0.000006	- 144

\*600nV is the Johnson Noise in a 10kHz BW of a 2.2k $\Omega$  Resistor @ 25°C

Remember: 10-bits and 10V FS yields an LSB of 10mV, 1000ppm, or 0.1%.  
All other values may be calculated by powers of 2.

**Figure 2.14: Quantization: The Size of a Least Significant Bit (LSB)**

# Základní vztahy pro převod

- Jaké číslo dostanu na výstupu pro napětí  $V_{in}$ ?

$$n = \left\lfloor \frac{(V_{in}) 2^N}{V_{REFSH}} + 1/2 \right\rfloor \quad \text{za předpokladu } V_{REFSL} = 0$$

↙ ořezání desetinné části – výsledkem má být celé číslo!

- Jaký interval napětí reprezentuje číslo  $n$ ?

$$V_{in\_min} = \frac{n - \frac{1}{2}}{2^N} (V_{REFSH})$$

$$V_{in\_max} = \frac{n + \frac{1}{2}}{2^N} (V_{REFSH})$$

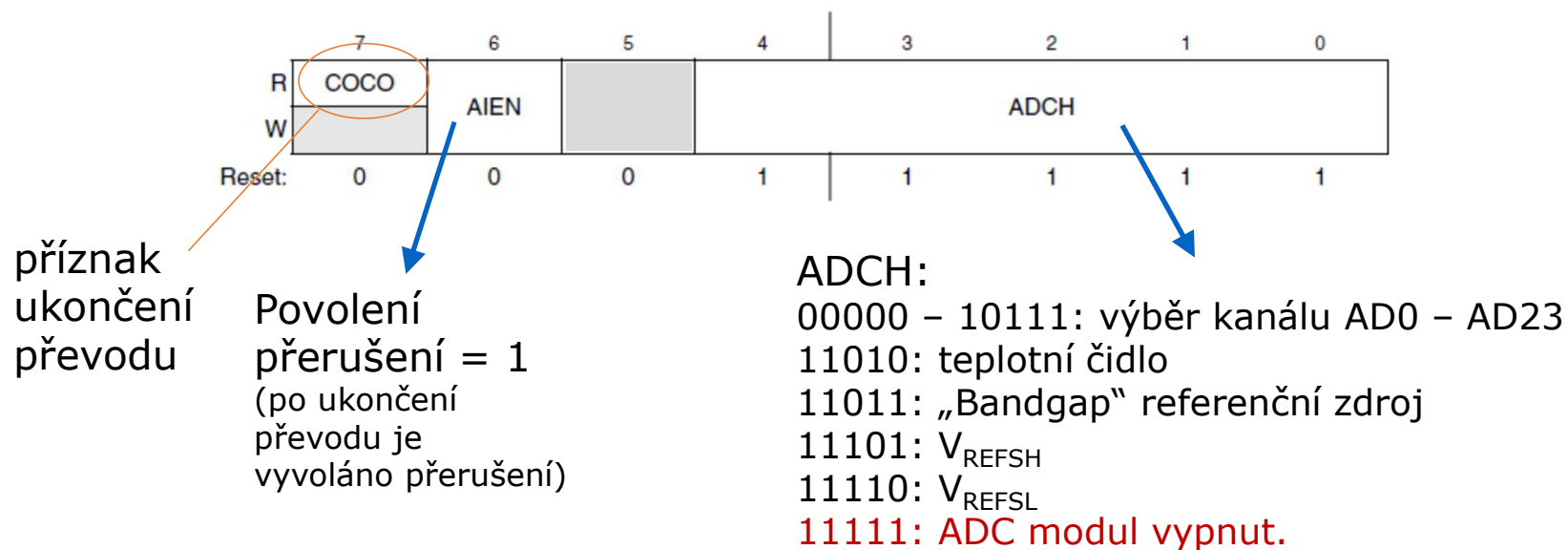
# Postup při převodu

- Převod aproximačním AD převodníkem je děj, který trvá určitou dobu. Proto lze vysledovat fáze:
  - zahájení převodu,
  - provádění převodu,
  - ukončení převodu (výsledek je k dispozici nebo byl převod „násilně“ přerušen).

# Zahájení převodu

- Zahájení převodu se děje zápisem do registru ADCx\_SC1A

(také existuje ADC0\_SC1B)

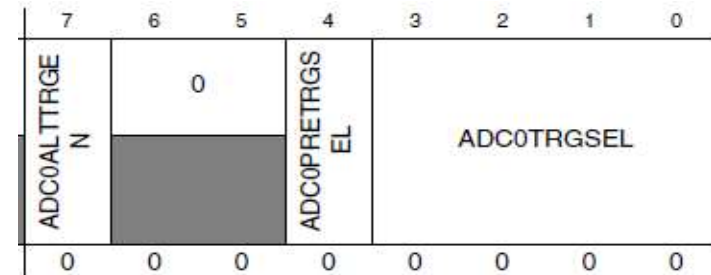


Registry ADCx\_SC1n jsou 32bitové, ale význam mají jen zde zobrazené bity!

# Zahájení převodu

- Jsou dvě možnosti spuštění převodu:
  - softwarové - zápis do registru ADCx\_SC1,
  - hardwarové – signálem ADHWT, který může být přiveden ze zdroje, voleného v registru SIM\_SOPT7:

3-0 ADC0TRGSEL	ADC0 trigger select
	Selects the ADC0 trigger source
0000	External trigger pin input
0001	HSCMP0 output
0010	Reserved
0011	Reserved
0100	PIT trigger 0
0101	PIT trigger 1
0110	Reserved
0111	Reserved
1000	TPM0 overflow
1001	TPM1 overflow
1010	TPM2 overflow
1011	Reserved
1100	RTC alarm
1101	RTC seconds
1110	LPTMR0 trigger
1111	Reserved



# Opakovaný převod

- Pokud je nastaven bit ADCO v registru ADCx\_SC3, pak se převod opakuje.
  - v případě softwarového spuštění se startuje další převod ihned po zápisu výsledku,
  - v případě hardwarového spuštění pak při další HW události (signál ADHWT).



# Výsledek

- Je k dispozici v registrech **ADCx\_Rn** - podle toho, kterým registrem byl převod zahájen (zápisem do ADC0\_SC1A se spustí převod, jehož výsledek bude v ADC0\_RA).
- Je indikován bitem **COCO** v registru ADCx\_SC1n.
- Může generovat přerušení.
- Lze nastavit, že výsledek je indikován jen když:
  - je menší než nějaká hodnota,
  - je větší nebo roven nějaké hodnotě
  - je v nějakém intervalu,
  - je mimo nějaký interval.

# Průměrování

- Modul ADC lze nastavit tak, aby provedl sám několik vzorkování a dodal jako výsledek **průměrnou hodnotu**.
- V registru ADCx\_SC3 jsou bity AVGE a AVGS, které toto nastavují:

AVGE	AVGS	Number of samples averaged
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

- Výsledek převodu (v registru ADC0\_Rn) je až průměrná hodnota.

# Režim porovnání

- Hodnota k porovnání se ukládá do registrů ADCx\_CV1 a/nebo ADCx\_CV2.
- Porovnání je možné
  - buď s hodnotou ADCx\_CV1 (výsledek převodu je větší nebo roven hodnotě výsledku) – bit ACREN v ADCx\_SC2 = 0
  - nebo s intervalem daným registry ADCx\_CV1 a ADCx\_CV2 (výsledek převodu je uvnitř/vně intervalu) – bit ACREN = 1.
- Porovnání se povoluje bitem ACFE v registru ADCx\_SC2.
- Způsob porovnání se nastavuje bitem ACFG T v registru ADCx\_SC2 (0 = menší, 1 = větší nebo rovno).
- Pokud bylo porovnání neúspěšné, výsledek převodu je ignorován: nezapíše se do datového registru, nenastaví se COCO, negeneruje se přerušování, ...

# Zrušení převodu

- Užitečné zejména při nastaveném opakovaném převádění, funguje ale i při jednorázovém převodu:
- Lze provést:
  - zápisem do ADC0\_SC1A (v tom případě se ale spustí nový převod, není-li ADCH = 11111),
  - zápisem do jiného registru ADC (mění se režim, tudíž právě probíhající převod nemá smysl dokončovat),
  - resetem MCU, vstupem do STOP režimu.

# Konfigurace - hodiny

- ADCx\_CFG1

- ADIV: nast. děličky hodin  $2^{ADIV}$

- 00: 1
- 01: 2
- 10: 4
- 11: 8

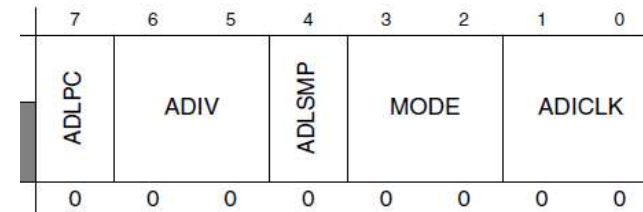
- ADICLK: Výběr zdroje hodin

- 00: Bus clock
- 01: Bus clock/2
- 10: ALTCLK
- 11: ADACK

- ADCx\_CFG2

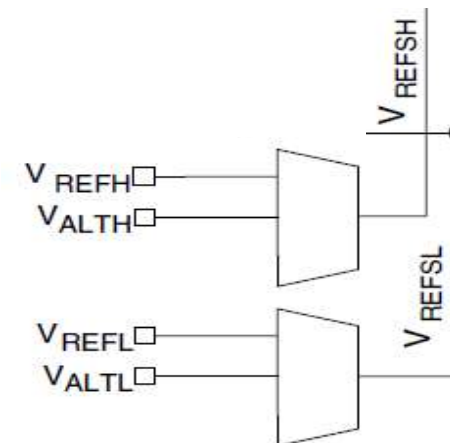
- ADACKEN: Povolení asynchronních hodin

Modul ADC má vlastní generátor hodin (asynchronní se zbytkem MCU).



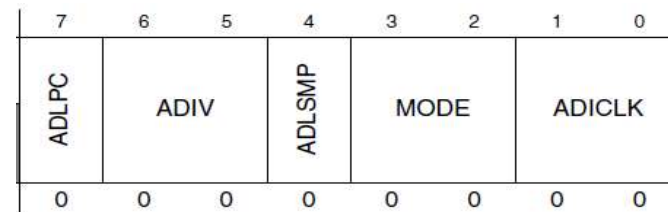
# Konfigurace – referenční napětí

- K dispozici jsou dvě referenční napětí:
  - napětí mezi  $V_{REFH}$ ,  $V_{REFL}$
  - napětí mezi  $V_{ALTH}$ ,  $V_{ATL}$
- Volí se v registru SC2 bity REFSEL
  - 00:  $V_{REFH}$ ,  $V_{REFL}$
  - 01:  $V_{ALTH}$ ,  $V_{ATL}$
  - 10, 11: Reserved
- „Náš“ MCU KL05Z má:
  - $V_{ALTH}$  connected to  $V_{DDA}$

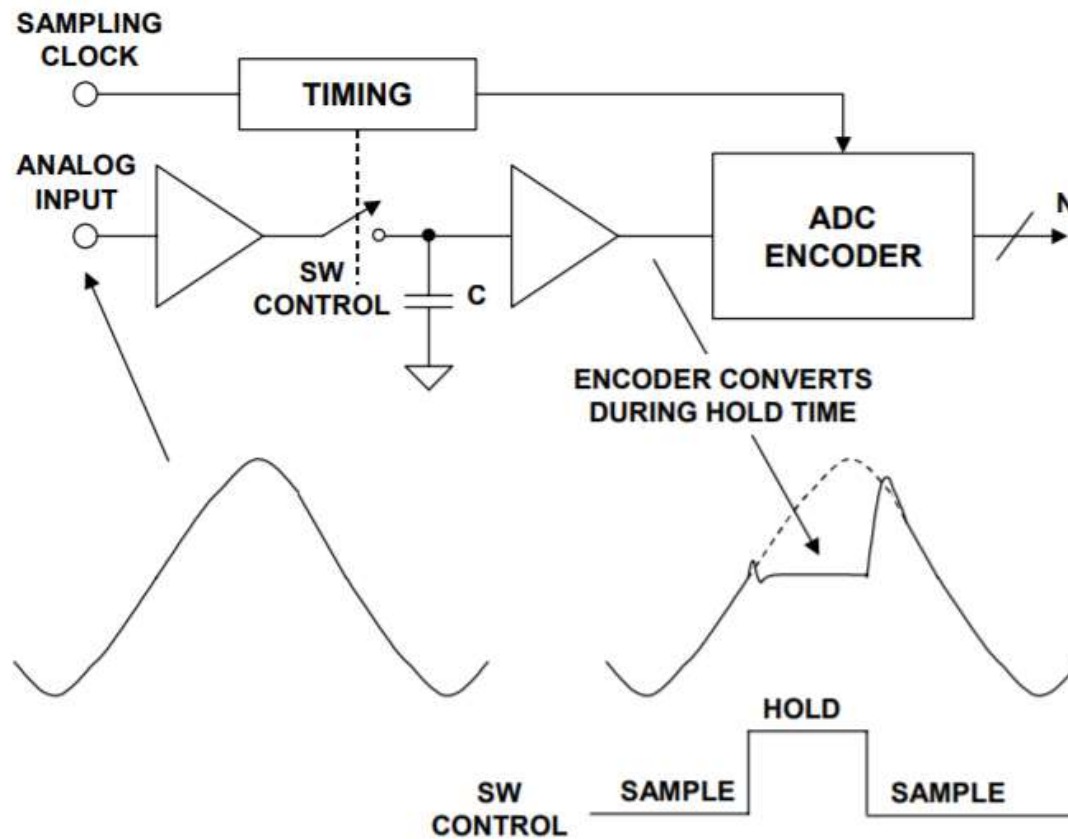


# Konfigurace – parametry převodu

- Low power
  - Bit ADLPC (v reg. ADCx\_CFG1) na 1
  - sníží rychlost hodin
- Long sample time
  - Bit ADLSMP (v reg. ADCx\_CFG1) na 1
  - je možné pomocí bitů ADLSTS (v reg. ADCx\_CFG2) přidat 20, 16, 10 or 6 ADCK cyklů)
- Conversion mode – režim převodu
  - Bity MODE (v reg. ADCx\_CFG1) nastavují 8, 10 nebo 12 bitů
- Continuous vs. single conversion
  - Set ADCO (in ADCx\_SC3) to 1 for continuous conversions



# Sampling time?



**Figure 2.27:** Sample-and-Hold Function Required for Digitizing AC Signals



# Jak dlouho trvá převod?

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder} + \text{HSCAdder})$$

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles <sup>1</sup>
1	0	11	5 μs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles <sup>1</sup>
0	0	11	5 μs + 5 ADCK cycles + 5 bus clock cycles

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
10b single-ended	20 ADCK cycles
12b single-ended	20 ADCK cycles

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

# Teplotní snímač

$$\text{Temp} = 25 - ((V_{\text{TEMP}} - V_{\text{TEMP25}}) \div m)$$

$$V_{\text{TEMP25}} = 1,396 \text{ V}$$

$$m = 3.266 \text{ mV/}^{\circ}\text{C pro rozsah -40 až 25 }^{\circ}\text{C}$$

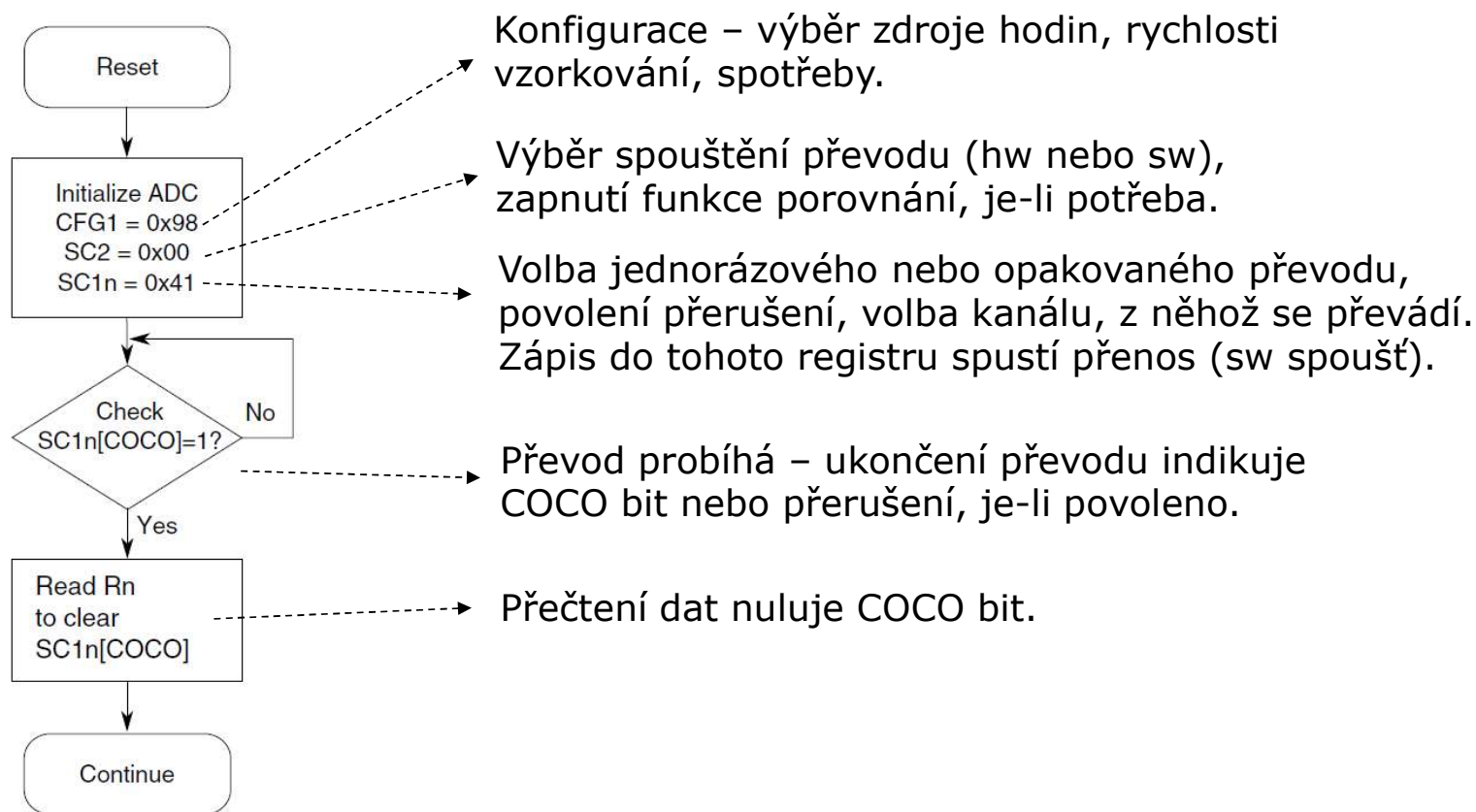
$$m = 3.638 \text{ mV/}^{\circ}\text{C pro rozsah 25 až 125 }^{\circ}\text{C}$$

Postup měření teploty:

- přečte se napětí kanálu AD26
- hodnota se porovná s  $V_{\text{TEMP25}}$
- je-li hodnota menší než  $V_{\text{TEMP25}}$ , do vzorce se dosadí první hodnota  $m$
- je-li hodnota větší než  $V_{\text{TEMP25}}$ , do vzorce se dosadí druhá hodnota  $m$

Zkalibrujeme-li teplotní snímač při 25°C, můžeme dosáhnout přesnosti až  $\pm 4,5^{\circ}\text{C}$ . Pokud kalibrujeme ve třech bodech (-40, 25 a 125°C), lze dosáhnout přesnosti až  $\pm 2,5^{\circ}\text{C}$ .

# Rekapitulace – jak se převádí



# ADC a spotřeba

ADC je energeticky náročný (postupná aproximace = spousta změn úrovní CMOS obvodů).

Je dobré zvážit,

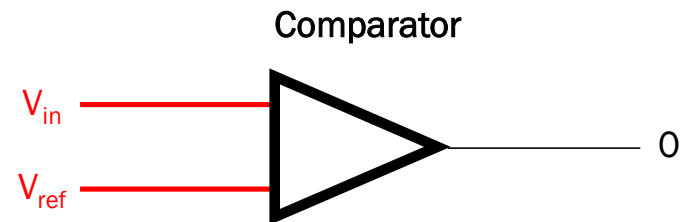
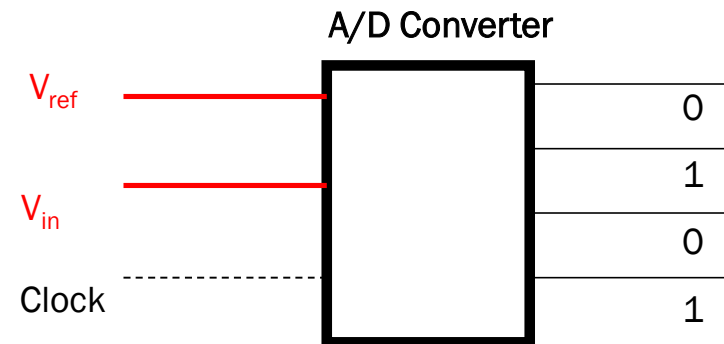
- **jak velkou rychlost převodu potřebujeme** – jak rychle chceme mít výsledek od okamžiku sběru vzorku signálu,
- **jak často potřebujeme vzorkovat,**
- **jakou přesnost potřebujeme** (každý bit výsledku stojí energii),
- **zda vůbec potřebujeme A/D**, když je k dispozici komparátor s desetinou energie ve srovnání s převodníkem, který v některých situacích vyhoví.

# Analogový komparátor

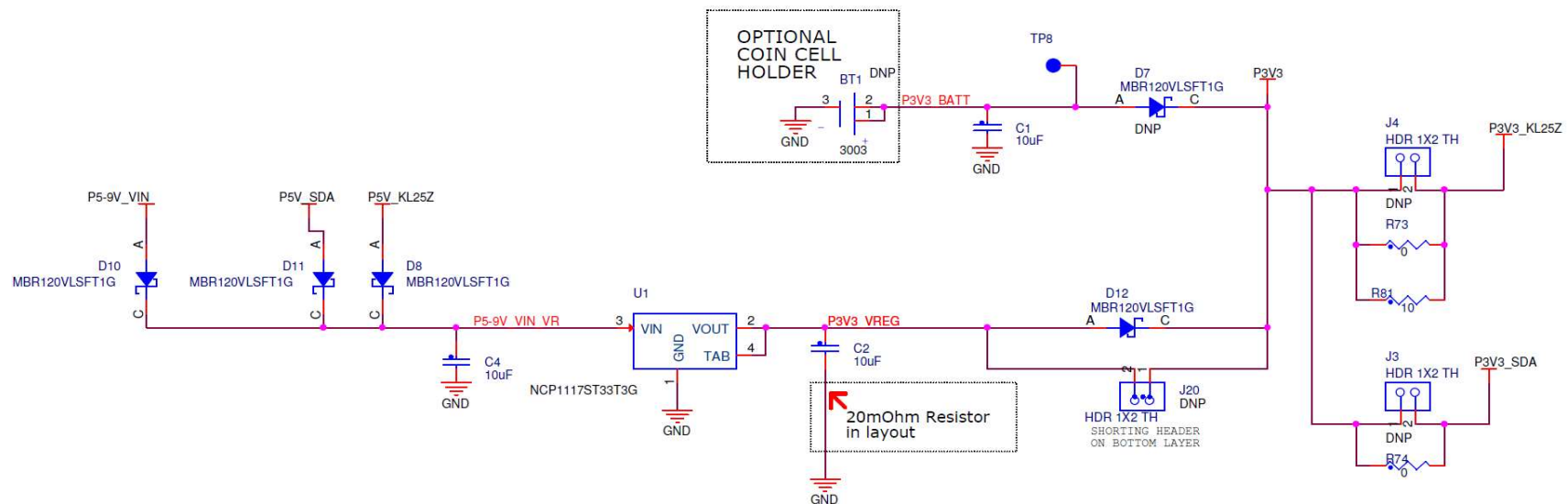
Modul CMP

# Analogový komparátor?

- A/D převodník (ADC) dává jako výstup číslo jak velké je  $V_{in}$  – vyjádřeno zlomkem  $V_{REF}$ .
- Analogový komparátor (CMP) říká, zda je  $V_{in}$  menší nebo větší než  $V_{REF}$  – binární hodnota.



## Příklad využití komparátoru – detekce ztráty napájení



Je třeba včas zjistit, že napájení vypadlo a v kondenzátoru C2 již zbývá jen trochu náboje, takže je třeba uložit kritická data, vypnout výstupy a přejít do bezpečného a úsporného režimu.

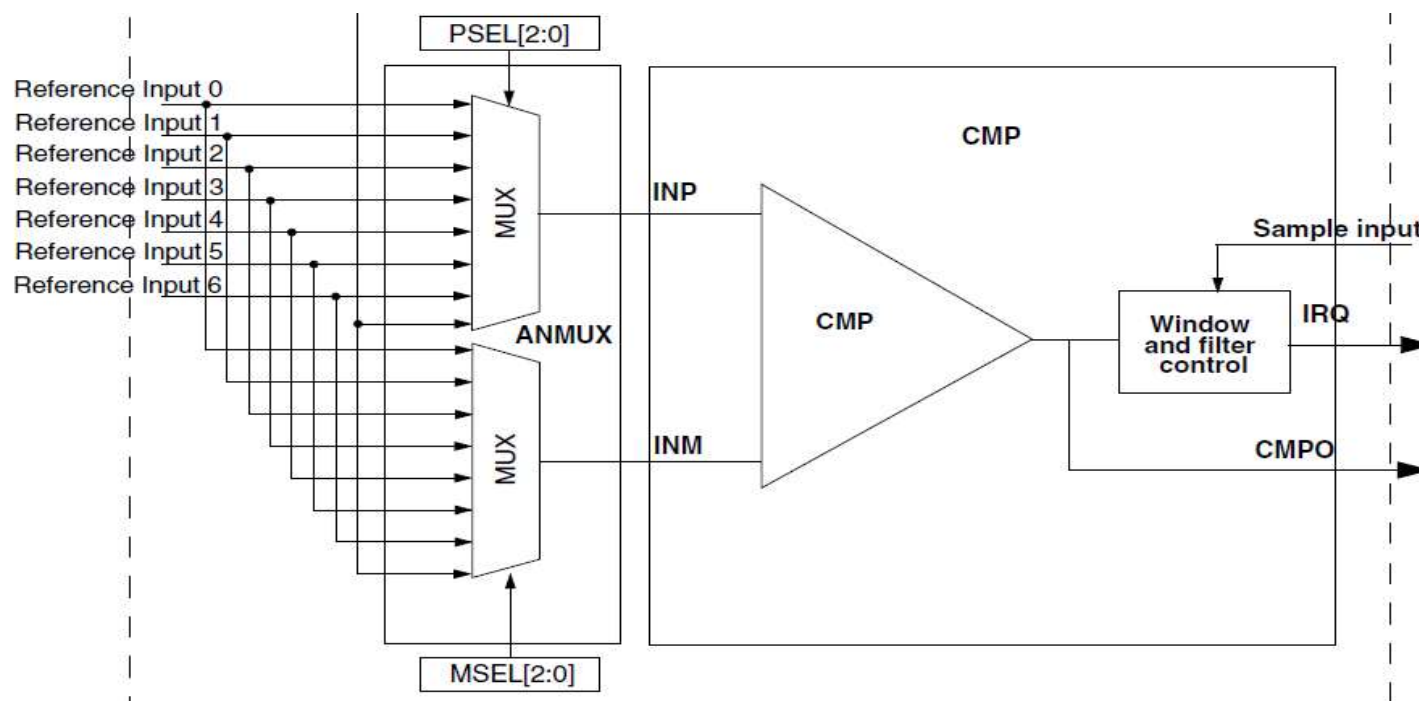
Můžeme využít komparátor pro srovnání napětí s referenčním zdrojem.

# Kdy je výhodný komparátor?

- Proč použít komparátor místo ADC?
  - komparátor má mnohem menší spotřebu,
  - výstup je aktuální vždy – je k dispozici okamžitě. Není třeba čekat na dokončení převodu.



# Schéma komparátoru CMP



Je-li  $INP > INM$ , pak výstup = 1, v opačném případě 0.  
Změna na výstupu může generovat přerušení.  
Porovnávané analogové hodnoty se vybírají multiplexery.

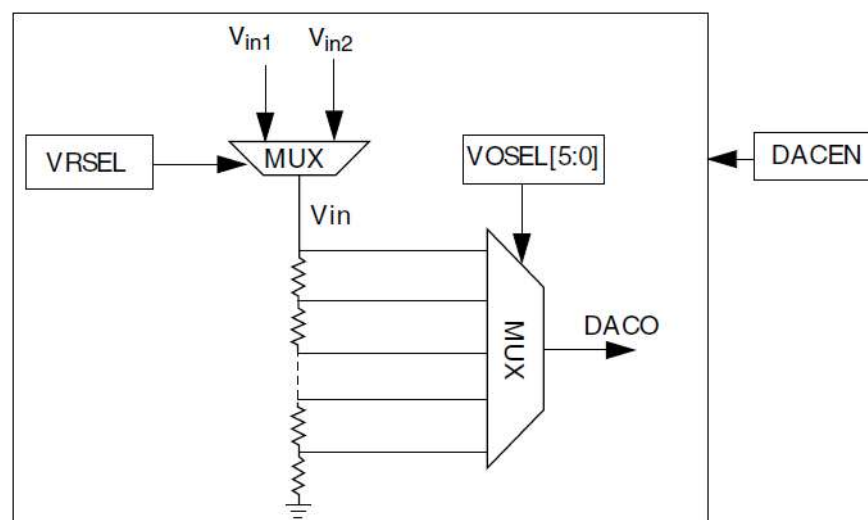
# Vstupy CMP

CMP Inputs	CMP0
IN0	CMP0_IN0
IN1	CMP0_IN1
IN2	CMP0_IN2
IN3	12-bit DAC0 reference/ CMP0_IN3
IN4	—
IN5	—
IN6	Bandgap <sup>1</sup> 1V referenční napětí
IN7	6-bit DAC0 reference

Vybírá se bity PSEL (vstup INP komparátoru) a MSEL (vstup INM) registru CMPx\_MUXCR:

Bit	7	6	5	4	3	2	1	0
Read	Reserved	0		PSEL			MSEL	
Write								
Reset	0	0	0	0	0	0	0	0

# 6-bitový DAC – možná reference pro CMP



Registr CMPx\_DACCR:

Bit	7	6	5	4	3	2	1	0
Read	DACEN	VRSEL	VOSEL					
Write								
Reset	0	0	0	0	0	0	0	0

Zapnutí

0 =  $V_{REFH}$   
1 =  $V_{DD}$

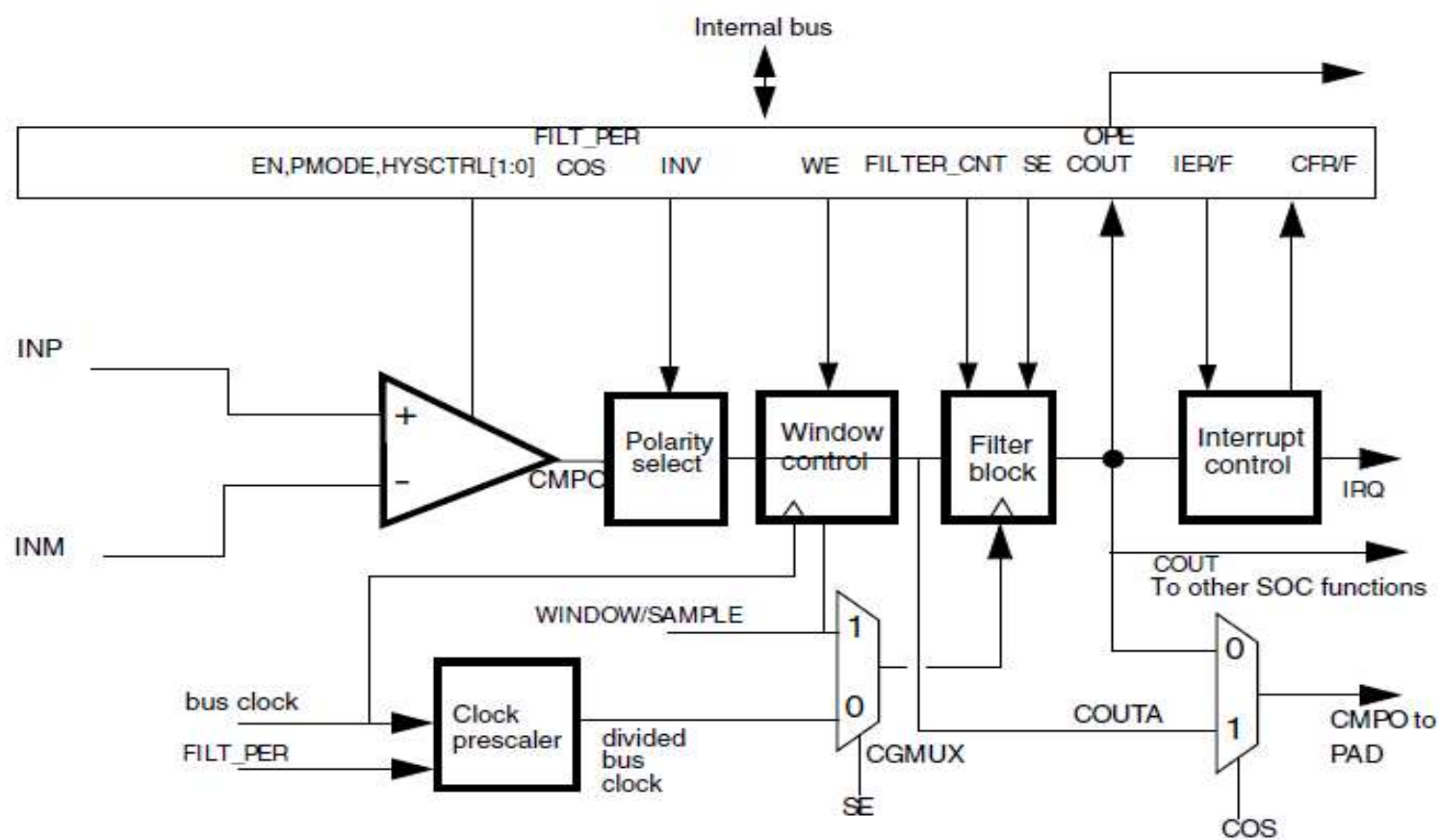
$$VOSEL = 64 * (V_{DACO} / V_{in}) - 1$$

$$V_{DACO} = (VOSEL + 1) * (V_{in} / 64)$$

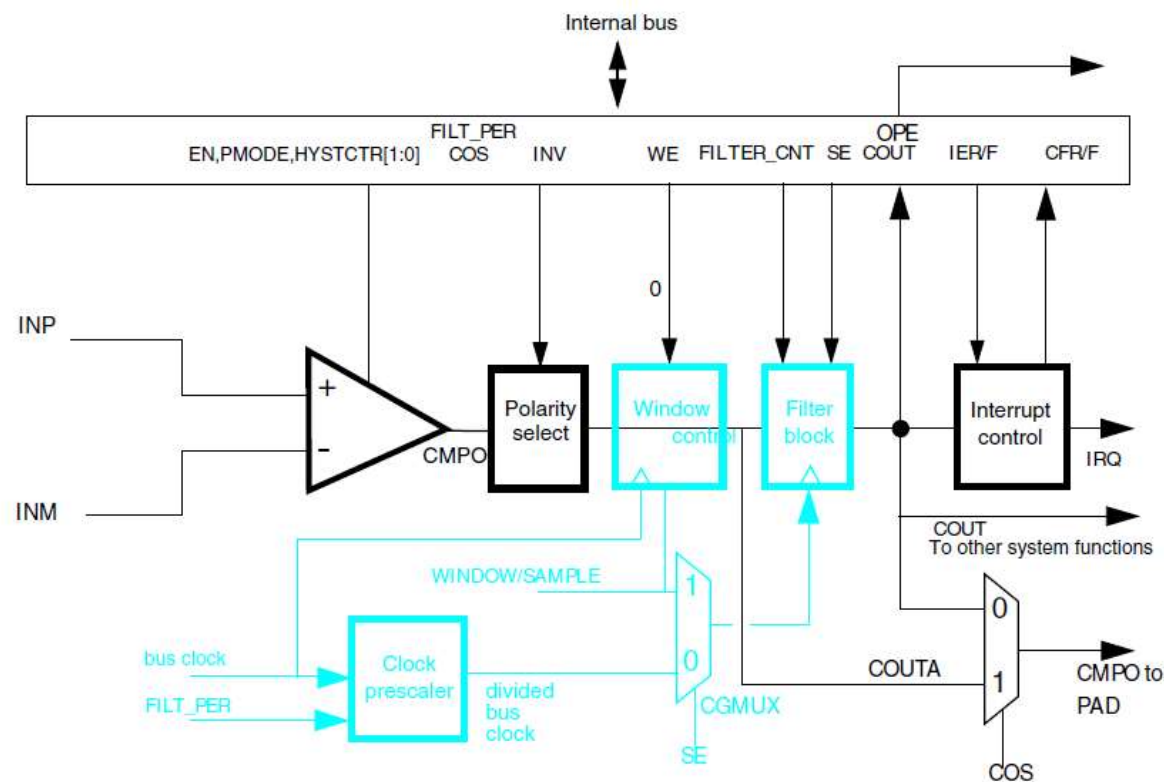
# Výstup CMP

- Zdánlivě jednoduchá věc – buď je tam 0 nebo 1, ale tak jednoduché to není – proč?
- Snímáme obvykle reálné veličiny v reálném prostředí!
- Výstup může kmitat v okamžiku, kdy je rozdíl mezi INP a INM velmi malý. Hraje roli šum, který může komparátor překlápět, rozkmitávat.
- Vstup je proto často třeba vhodně filtrovat.

# Výstup CMP

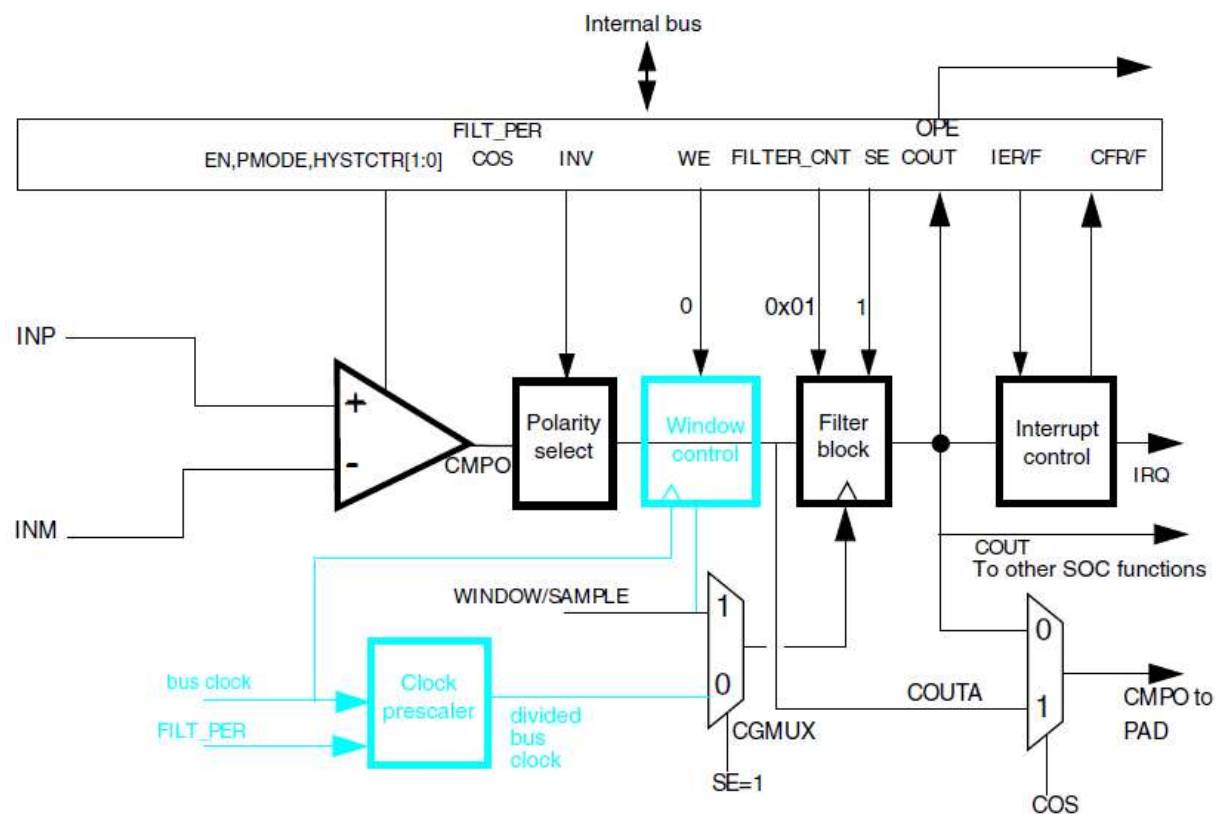


# Výstup CMP – continuous mode



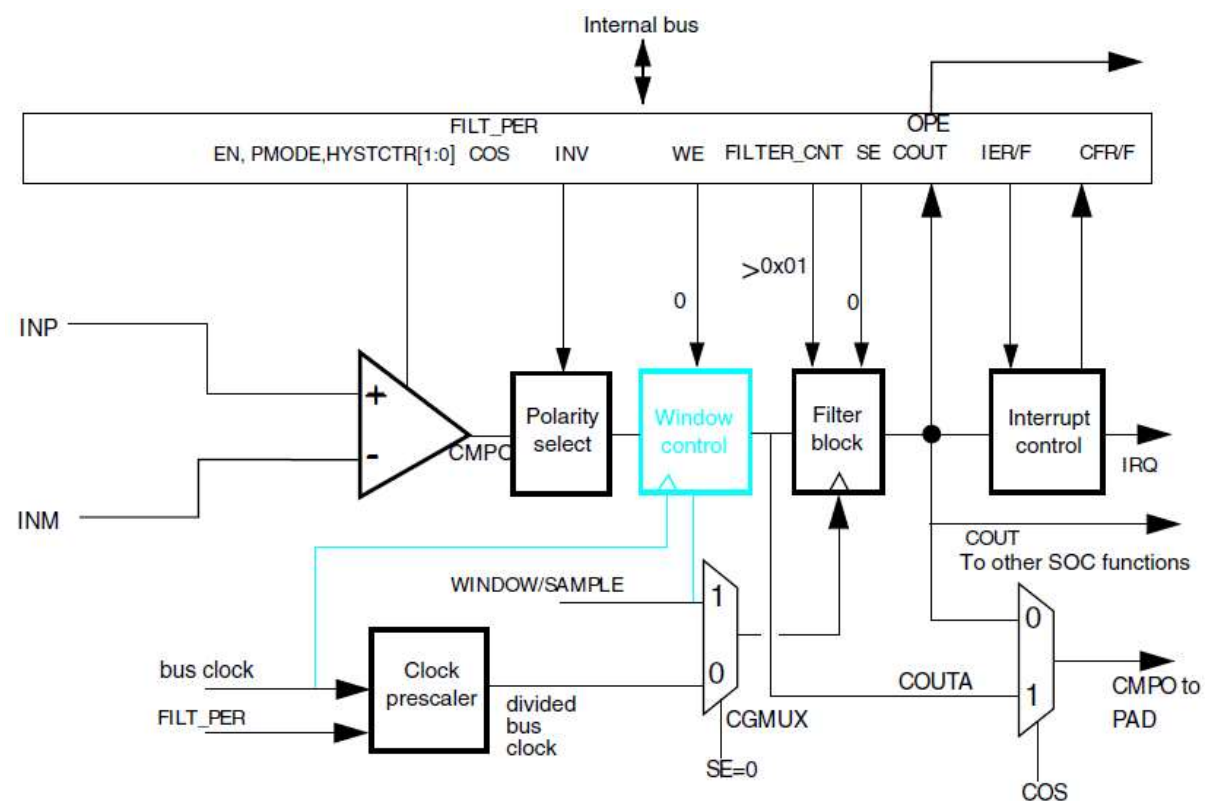
Římý výstup – není nijak filtrován ani vzorkován.

## Výstup CMP – sampled, non-filtered



Výstup je vzorkován hodinami, lze zvolit jakými – zde externími.

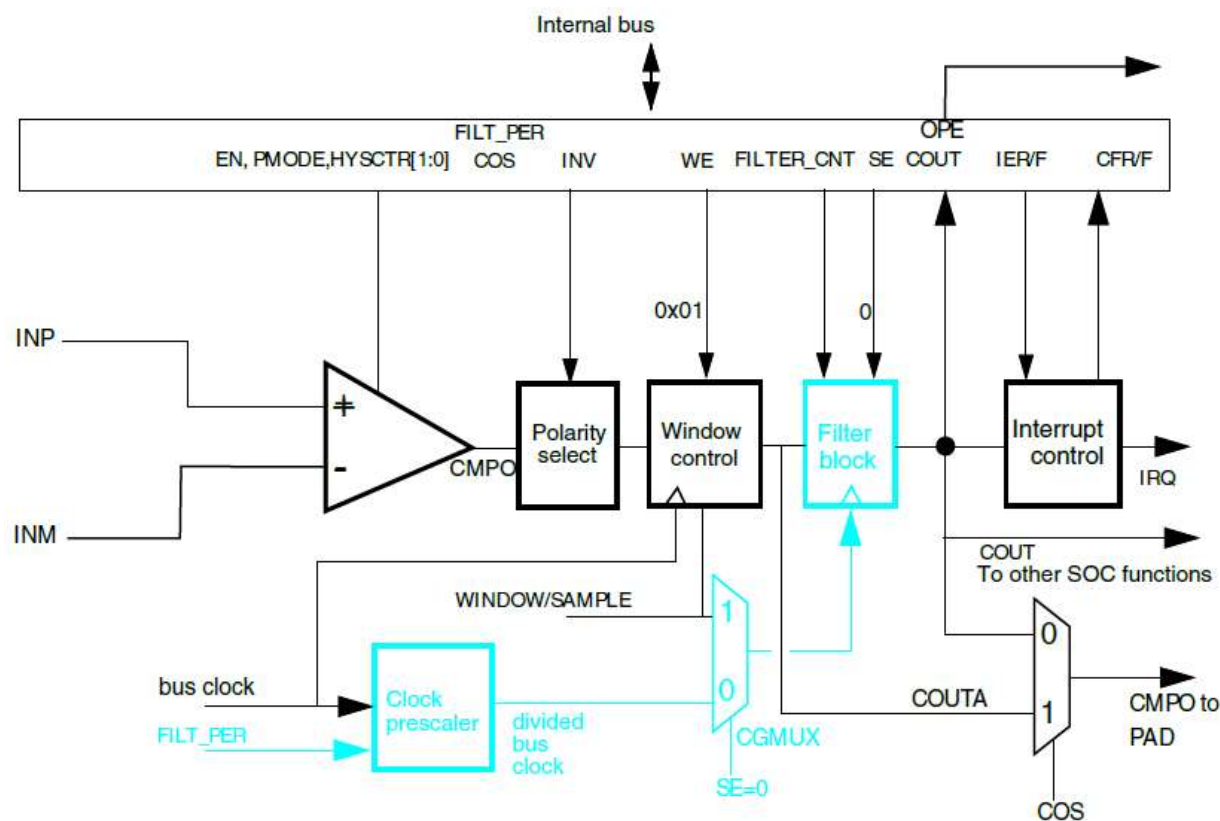
## Výstup CMP – sampled, filtered



Výstup je vzorkován hodinami, lze zvolit jakými – zde interními.  
Navíc je zapnut filtr – hodnota `FILTER_CNT` říká, kolik vzorků se musí shodnout.

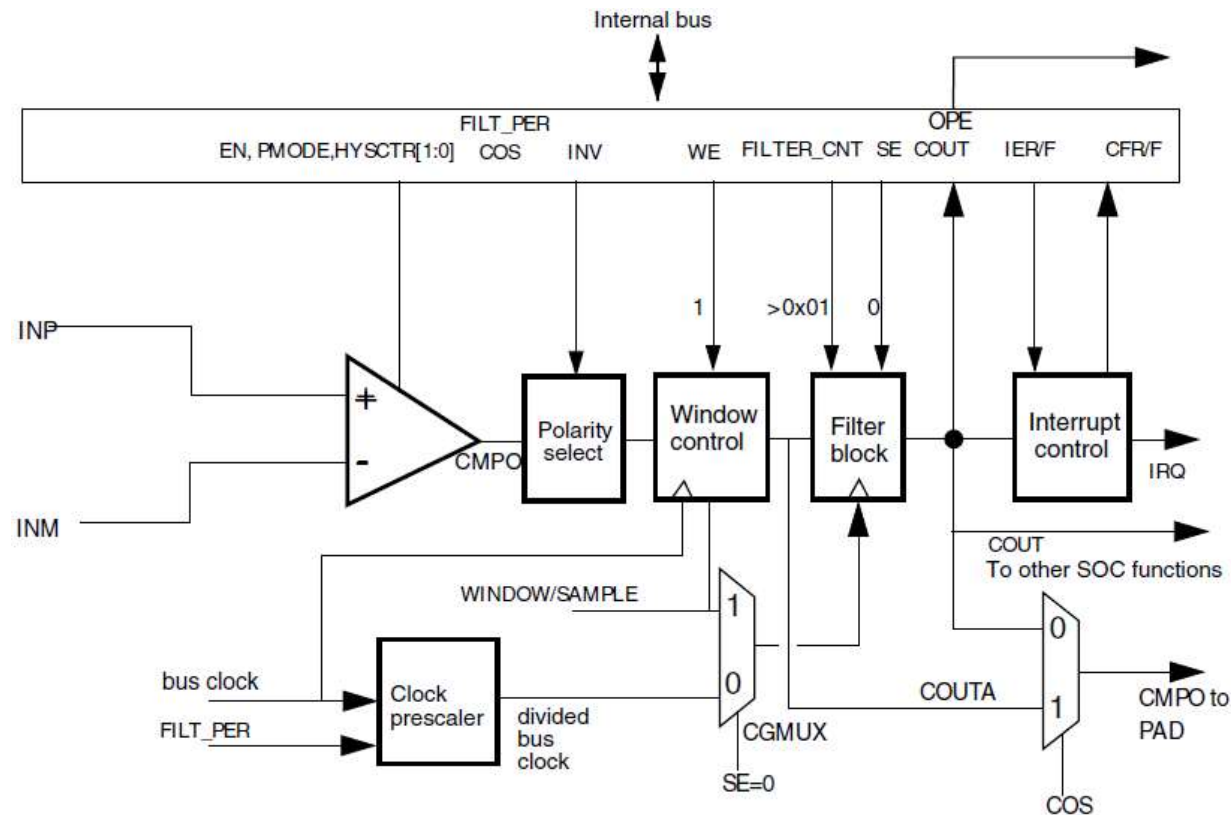


# Výstup CMP – windowed mode



Výstup je brán v potaz jen pokud je aktivní signál WINDOW.

## Výstup CMP – windowed, filtered

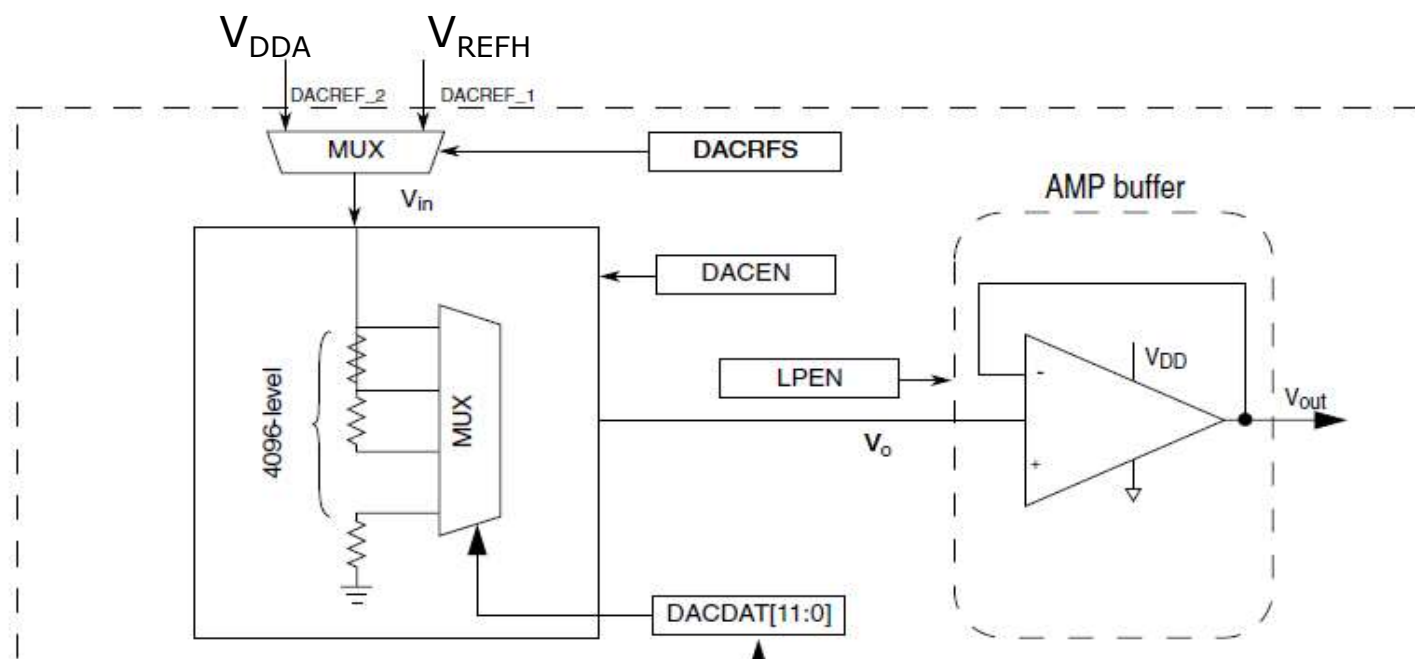


Výstup je brán v potaz jen pokud je aktivní signál WINDOW.  
Navíc je zapnut filtr – hodnota FILTER\_CNT říká, kolik vzorků se musí shodnout.

# Číslicově-analogový převodník

modul DAC

# DA převodník – modul DAC



$$V_o = (N+1) * V_{in} / 2^{12}, \text{ kde } N \text{ je 12-ti bitové číslo } DACDAT[11:0]$$

Prosím, nezaměňujte modul DAC s D/A převodníkem v modulu CMP.

# Režimy činnosti

- V normálním (bez bufferu) režimu DAC převádí číslo z registrů DACDAT0 (dva osmibitové registry).
- V bufferovaném režimu lze převádět postupně čísla z 16ti místného bufferu, další vzorek se převede buď
  - zápisem 1 na bit DACSWTRG v registru DACxC0 nebo
  - hw triggerem (od PIT).
- buffer se chová buď
  - jako kruhový nebo
  - se projde jednou a pak se zastaví.

# Pro úplnost - PWM

- (Pseudo)analogový výstup lze vytvořit také pomocí PWM (viz přednáška o časovačích).

