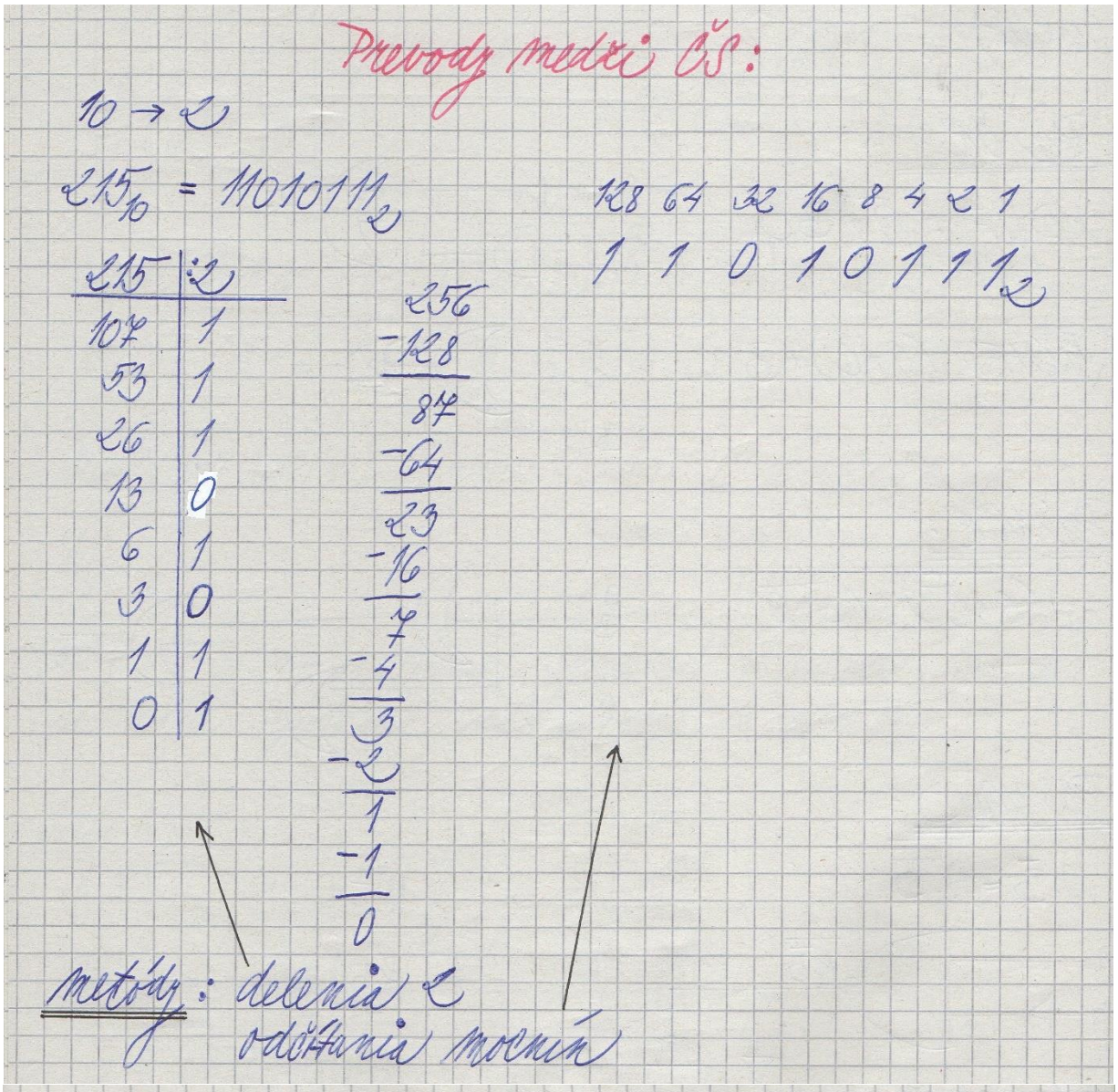
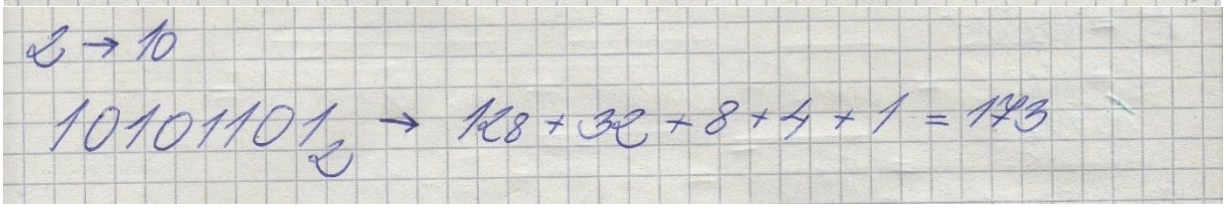
3. Jednočipový mikropočítač

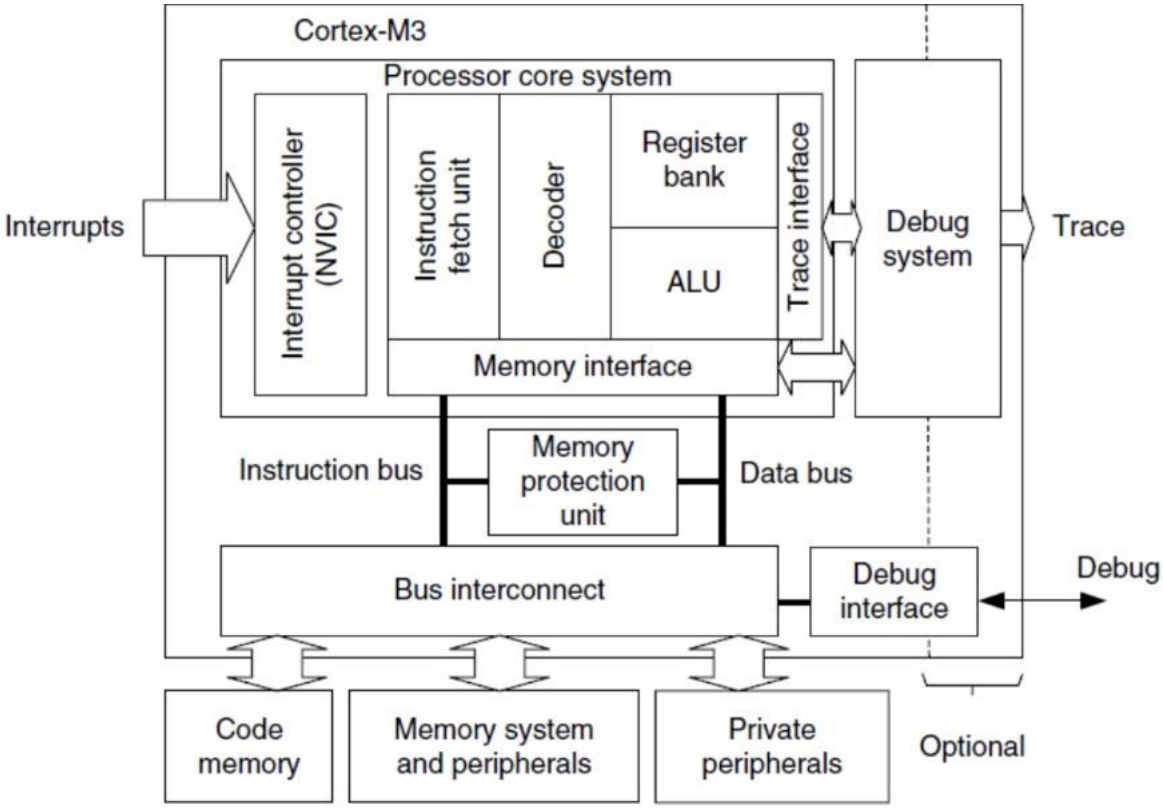
- **Prevody medzi číselnými sústavami:** *napr.* **2→10** a **10→2**

Čísla ako také delíme na ⎧ **POZIČNÉ** - $1299_{10} = 10^3\ 10^2\ 10^1\ 10^0$
⎩ **NEPOZIČNÉ** - LXX





ARM Cortex-M



Cortex-M3 obsahuje: *// vedieť na maturite popísať obrázok*

-radič prerušení (NVIC - Nested Vectored Interrupt Controller) – riadi prerušovací systém procesora (prerušovací system si vysvetlíme neskôr)
-jednotku načítania inštrukcií (Instruction fetch unit) – vykonáva fetch state
dekodér inštrukcií – vykonáva decode state
-registre
-aritmeticko-logickú jednotku (ALU) – ALU vykonáva aritmeticko-logické operácie nad dátami uloženými v registroch – vykonáva execute state
-pamäťové rozhranie (Memory interface) – slúži na komunikáciu jadra procesora s pamäťou (zabezpečuje, aby si jadro procesora s pamäťou rozumeli)
-rozhranie pre krokovanie programu (Trace interface)
-podporu a rozhranie pre funkcie ladenia programu (Debug interface)
-voliteľnú jednotku ochrany pamäte (Memory protection unit)

MikroCPU = súčiastka, CPU

MikroPC = PC, ktorý používa mikrokontrolér, ako svoj CPU

Inštrukcia = najjednoduchší úkon CPU

1. Koľko má bitov a čo tento údaj znamená?

*Cortex-M3 je **32 bitový procesor**, t.j. pracuje s 32 bitovými údajmi (32-bit slovom)*

Maturitné otázky, odbor-POS

2. Koľko bytová je adresa? Čo to znamená?

*Má **32-bytovú adresnú zbernicu**, t.j. **2^{32} adries = 2^{32} B**. Je schopný používať **4GB adresovateľného pamäťového priestoru**.*

3. Akú má architektúru?

Harvardskú** – pamäť je **rozdelená** → pamäť pre **program** (kód) a pre **dáta

4. Akým spôsobom sa spracovávajú Inštrukcie + vysvetlenie

*Využíva **3-stupňové prúdové spracovanie inštrukcií PIPELINE** → vďaka nemu je schopný vykonávať viac inštrukcií súčasne.*

Ak sa 1. inštrukcia vykonáva (EXECUTE), nasledujúca sa DEKÓDUJE a ďalšia sa vyberá/načítava z operačnej pamäte do RI (FETCH)

5. V akých módoch/režimoch pracuje procesor ARM Cortex-M? (Módy, ktoré nie sú naviazané na zásobník) + ich charakterizovať

***THREAD** – pokiaľ sa vykonáva inštrukcia bez prerušenia, ale keď nastane výnimka / prerušenie, tak sa prepne do:*

***HANDLER** – spracováva sa výnimka, prerušenia + výnimky*

6. V akých režimoch pracuje ARM Cortex M3 + charakterizovať tieto režimy (tie, ktoré sú naviazané na vlastný zásobník) + ku každému pripísať zásobník

***PRIVILEGOVANÝ** (má plnú kontrolu) – používa **MAIN STACK** (zásobník), CPU má prístup k celému setu inštrukcií*

***NEPRIVILEGOVANÝ** (niečo má obmedzené) – má len určité fcie, registre obmedzené, pracuje s **PROCESS STACK** (zásobník)*

Zásobník:

Špeciálny druh pamäte, ktorý je použitý na dočasné uloženie informácií

Slúži aj na to, keď chceme čítať informácie v opačnom smere, ako boli zapísané do pamäte (nemusíme riešiť adresovanie údajov)

Je to sekvenčná pamäť, umožňuje zápis info do/z 1 MIESTA (horná časť zásobníka)

Keď zapisujeme informáciu, tak sa posúvajú zhora na ďalšie miesto smer nadol (hĺbka)

Načítať vieme len zhora! → odstránia sa tak INFO len z vrchnej časti, ďalšie tam prídu

Poradie typu LIFO

Na manipuláciu so zásobníkom využijeme pokyny:

1. PUSH – ZÁPIS (vloží do zásobníka nový údaj, vložením na vrch)

2. READ – ČÍTAŤ hornú časť Z

3. POP – ODSTRÁNI údaj z hornej časti a posunie obsah nahor

Maturitné otázky, odbor-POS

7. Do akého režimu prechádza ARM Cortex M3 po reštarte / resete?

Začína v THREAD MODE, PRIVILEGOVANOM REŽIME → ten pracuje s Main stack

8. Vysvetlenie prerušenia

Žiada o neho HARDVÉR, aby PROCESOR prerušil vykonávanie programu a obslúžil požiadavku zariadenia, potom sa vráti tam kde bol prerušený v pôvodnom programe (túto informáciu si predtým uloží do zásobníka)

9. Prerušovací systém:

Súbor všetkých prerušení → každé periférne zariadenie, ktoré chce niečo od procesora, ten preruší vykonávanie programu a obslúži požiadavku. Každá táto požiadavka má inú prioritu a každá má svoje Číslo a spôsob ako ju CPU vykoná. Existuje vektorová tabuľka výnimiek, kde prvých 15 sú výnimky prerušení

Prerušenie prerušenia:

*Príde požiadavka na prerušenie s **vyššou prioritou v čase vykonávania prerušenia s nižšou prioritou**. Obsluha prerušenia s nižšou prioritou sa zastaví (CPU si uloží do pamäte = zásobníka informáciu o tom kde skončil vykonávanie programu). Následne začne spracovávať požiadavku s vyššou prioritou, ktorá prerušila vykonávanie. Keď skončí vykonávanie požiadavky, tak načíta informáciu zo zásobníka, tam kde skončil obsluhu požiadavky s nižšou prioritou a pokračuje v jej vykonávaní.*

10. Čo je to BUS Matrix:

*Je to **sieť vysokorýchlostných zberníc**, zbernica ktorá umožňuje paralelné cesty medzi Cortex zbernicami a ďalšími externými zbernicami AHB (Advanced High Speed Busses)*

11. Vysvetlenie systémového časovača:

Poskytuje ho 24 bit počítadlo impulzov, na ktorého vstup privedieme hodinový signál z oscilátora → počíta dozadu

12. Použitie SysTick:

Na určenie systémového taktu pre RTOS (Real Time Operation System), na generovanie periodických prerušení pre plánované úlohy, má 3 Registre

13. BIT Banding → technológia na urýchlenie manipulácie s bitmi. Ak chceme zmeniť len 1 bit v údaji: procesy (načítania, maskovania, modifikácie, zápisu) sú zdĺhavé

14. Z akých typov môže byť spustený program?

Code, SRAM, Peripheral, External RAM

15. Tail chaining: metóda koncového zariadenia, má za úlohu minimalizovať oneskorenie (trvá 6 cyklov)

*16. Registre CPU; **Register = pamäť CPU***

- RO-R12 (jednoduché registre)

- R13 – Stack pointer (ukazovateľ zásobníka) → umožňuje CPU pracovať v 2 módoch (režimoch) → MAIN a PROCESS STACK

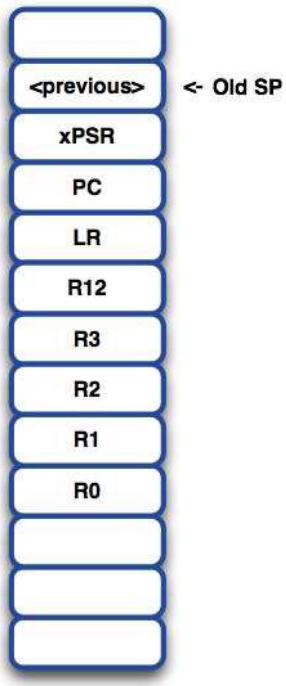
- R14 – link register (na uloženie návratovej adresy pri volaní podprogramu)

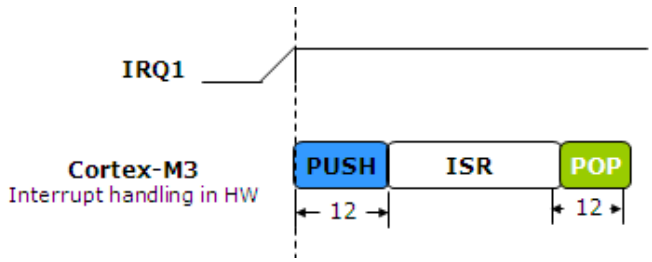
- R15 – program counter

Začiatok a koniec obsluhy prerušenia *// vedieť popísať obrázky*

Keď príde požiadavka na prerušenie od periférie, NVIC naštartuje CORTEX-M CPU na proces obsluhy prerušenia. Z dôvodu, že CORTEX-M CPU prechádza do režimu prerušenia, je potrebné uchovať informácie o prerušenom programe. Obsah určených registrov sa uloží do zásobníka (PSP alebo MSP). Keď je obsah registrov uložený do zásobníka, vyberie sa cez inštrukčnú zbernicu adresa podprogramu na obsluhu prerušenia.

Čas od požiadavky na prerušenie po prvú inštrukciu obslužného podprogramu prerušenia je **12 cyklov**, počas ktorých procesor zrealizuje vyššie uvedené úkony.



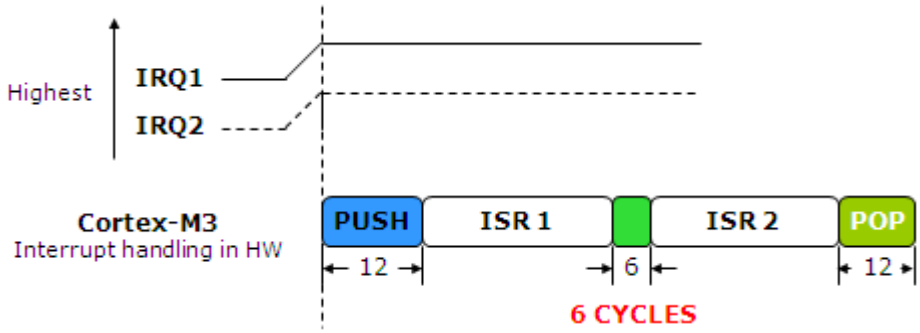


Obrázok 16: Počet cyklov začiatku a ukončenia obsluhy prerušenia

Obrázok 17: Registre, ktoré sa pri obsluhe prerušenia ukladajú do zásobníka

Metóda koncového zreťazenia (Tail Chaining)

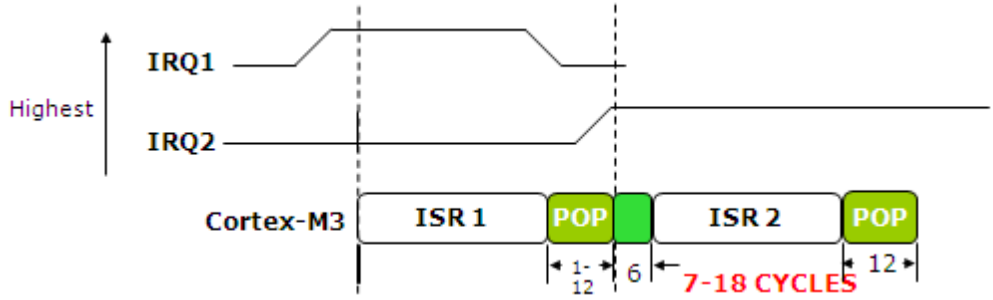
Ak beží obsluha prerušenia s vyššou prioritou a príde požiadavka na prerušenie s nižšou prioritou, použije sa metóda Tail Chaining, aby sa zabezpečilo minimálne oneskorenie medzi obslužnými podprogramami .



Obrázok 18: Tail Chaining

Ako vidieť z obrázka, počas 12 cyklov sa uložia registre do zásobníka, vyberie sa adresa obslužného podprogramu. Po ukončení obslužného podprogramu sa neobnovia registre zo zásobníka, ale načíta sa adresa obslužného podprogramu ďalšieho prerušenia podľa priority – 6 cyklov. Po ukončení obsluhy prerušenia sa obnovia registre zo zásobníka – 12 cyklov.

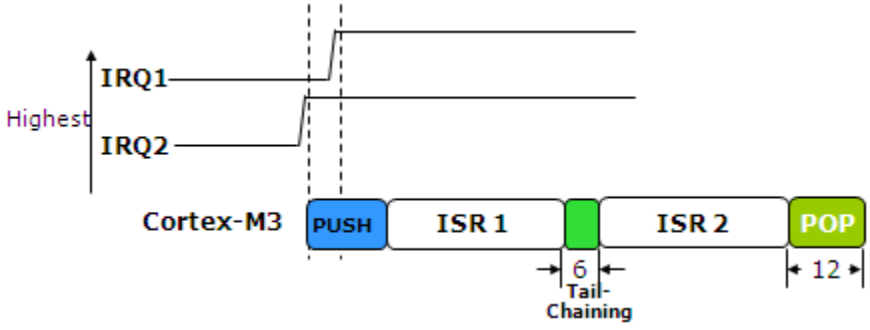
Ak počas obsluhy prerušenia príde požiadavka na prerušenie s nižšou prioritou, od obnovovania registrov bude upustené, ukazovateľ zásobníka sa obnoví na pôvodnú hodnotu, pridá sa 6 cyklov na vybratie adresy obslužného podprogramu prerušenia s nižšou prioritou t.j. oneskorenie je 7 – 18 cyklov pred obsluhou nového prerušenia. (viď. nasledujúci obrázok)



Obrázok 19: Reťazenie obsluhy prerušení

Oneskorený príchod požiadavky s vyššou prioritou

V realite sa často stáva, že začne obsluha požiadavky na prerušenie s nižšou prioritou a príde následne požiadavka s vyššou prioritou. Pokiaľ sa tak stane v inicializačnom ukladaní registrov do zásobníka (PUSH), prepne sa na obsluhu prerušenia s vyššou prioritou, po jej ukončení nasleduje Tail Chaining.



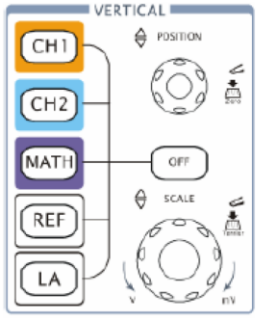
Obrázok 20: Oneskorený príchod požiadavky na prerušenie s vyššou prioritou

Vysvetlite činnosť vertikálneho kanálu a vzorkovania v DO:

Zobrazuje veľkosť napätia na **y osi**. Vzdialenosť od osi x nám určuje **amplitúdu** meraného signálu. Merací rozsah ***[V/diel]***

Vertikálny systém *obr. 8*
zobrazuje vertikálne ovládanie, tlačidlá ***CH1, CH2, MATH, REF*** a ***OFF*** a vertikálne ovládače ***POSITION*** a ***SCALE***.
POSITION – vycentruje signál na displeji, behom otáčania ovládača sa krátko zobrazí hodnota napätia vo vzťahu k zemnému potenciálu umiestnenom v strede obrazovky.
Tlačidlo rýchleho návratu vertikálneho ofsetu na „0“ – ak stlačíme tlačidlo ***POSITION.***
SCALE – prepínanie hrubého/jemného nastavenia pomocou stlačenia tohto tlačidla.
Stavový riadok sa nachádza na spodnej časti obrazovky a pomocou neho získavame informácie o meraní. Stlačením tlačidla ***OFF*** sa stavový kanál vypne.

obr. 8 Okno vertikálneho systému



Pulzne kódová modulácia = Používa sa na - prevod z analógového signál na digitálny

Akou zmenou prechádza analógový signál?

1. Vzorkovanie:

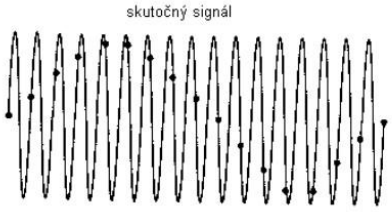
v určitom časovom okamihu odoberáme z analógového signálu **vzorky**, ktoré zodpovedajú **napäťovej hladine** → zápis do **akvizačnej pamäte** typu *FIFO*

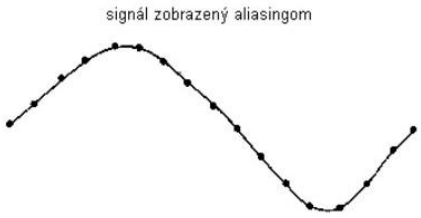
Pri vzorkovaní platí **SHANON-KOTEĽNÍKOV TEOREM**
Vzorkovacia frekvencia je min 2x väčšia, ako max. frekvencia signálu

$f_{vzork} = 2 * f_{max}$

$\Delta t = \frac{1}{2 * f_{max}}$ → čas spustenia ďalšej vzorky

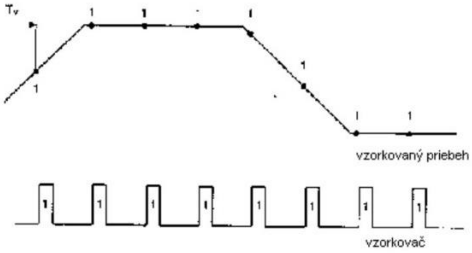
Je tu možný výskyt **aliasing** efektu (falošne kópie) – vieme ho obísť prekladaním vzorkovaním, t.j. budeme viackrát vzorkovať pri sebe

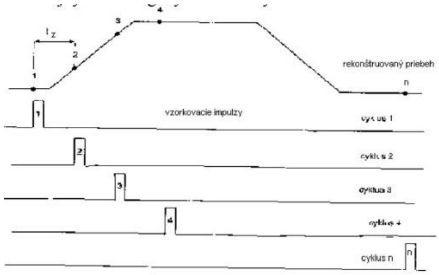


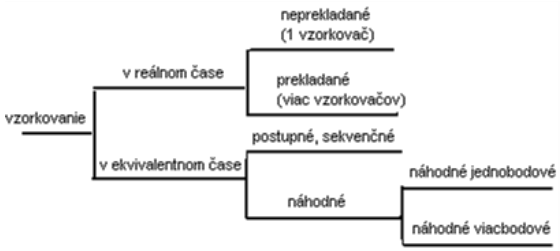


Aliasing *objavíme skusmým prepnutím* ***časovej základne*** *na rýchlejší rozsah a tak nájdeme* ***správny*** *obraz priebehu. Keď zveríme nastavenie osciloskopu* ***automatickému*** *nastaveniu (**AUTOSET**, **AUTOSCALE**), je pri jeho činnosti rozpoznaná* ***skutočná frekvencia*** *pozorovaného signálu a automatické nastavenie zvolí* ***správnu časovú základňu***

Reálne vzorkovanie - Celý signál sa uloží do Pa DO a naraz sa odoberú vzorky
Ekvivalentne vzorkovanie – Celý signál sa načíta do Pa a odoberie sa jedna vzorka, opäť sa načíta do Pa a odoberie sa ďalšia vzorka







Vlastnosti jednotlivých spôsobov vzorkovania :

metóda	výhody	nevýhody
vzorkovanie v reálnom čase	• zobrazenie jednorazových dejov • aliasing	• obmedzuje frekvenčný rozsah rýchlosť vzorkovania • nie je možné použiť priemerovanie, obálku a detekciu špičiek
vzorkovanie v ekvivalentnom čase	• zvýši frekvenčný rozsah pre opakovaný signál • možnosť priemerovania obálky a detekcia špičiek	• vyžaduje opakovaný signál • aliasing

Vysvetlite druhy platobných kariet, porovnajte ich použitie v praxi

Základné rozdelenie platobných kariet –

1. debetná – pri platení používate vlastné peniaze, výber z bankomatu a platbu je potrebné potvrdiť zadaním PIN kódu, prípadne bezkontaktne

2. kreditná karta – úverová karta, teda pri platení používate peniaze banky. Úver je revolvingový, primárne určený na bezhotovostné platby

3. charge karta – úverová karta s prideleným úverovým rámcom, bezúročné obdobie môže byť aj 60 dní

Základné bezpečnostné pravidlá pre používanie platobných kariet sú na webových stránkach jednotlivých bánk.