-8. 9. 2023

# Opakování

## Převody do 10 soustavy

37,0625Desítková =10 0101, 0001Hexa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 37/2 | 18/2 | 9/2 | 4/2 | 2/2 | ½ |
| 18 zb. 1 | 9 zb. 0 | 4 zb. 1 | 2 zb. 0 | 1 zb. 0 | 0 zb. 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0,0625 \* 2 | 0,125 \* 2 | 0,25 \* 2 | 0,5 \* 2 |
| 0,125 | 0,25 | 0,5 | 1 |

47,21Desítková =H 2F, 35C

|  |  |
| --- | --- |
| 47/16 | 2/16 |
| 2 zb. 15 | 0 zb. 2 |

|  |  |  |
| --- | --- | --- |
| 0,21 \* 16 | 0,36 \* 16 | 0,76 \* 16 |
| 3,36 | 5,76 | 12,16 |

B – H

1011101, 110111  
0101 1101, 1101 1100  
5D,DC

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | 4 | 2 | 1 |

H – B

3B,4E6  
0011 1011, 0100 1110 0110

# Aritmetické operace v číselných soustavách

10+10=20

1B+1B=10B – Aritmetický součet

1+1=1 – Logický součet OR

## Součet

Nulová 27 + 15 = 42

Hexa 2F + 1C = 4B  
 F + C = 27 = 1\*16 + 11

13. 9. 2023

# Sčítání v binární soustavě

## Způsob

1101 1 + 1 = 2 = 1 \* 2 + 0  
 110 1 + 1 + 1 = 3 = 1 \* 2 + 1  
1110 = 5 = 2 \* 2 + 1  
 101

## Způsob

11011001 + 10011101 = 101110110

## BCD Kód

* Binárně dekadický kód každou číslici zapíše pomocí 4 bitů dvojkového čísla
* Největší číslo je 9
* Například 1001BCD; 1 0D = 0001 0000BCD

## Součet v BCD kódu

657D + 248D = 905D

0110 0101 0111BCD + 0010 0100 1000BCD = 1000 1001 1111BCD  
když je číslo větší než 9 přičteme 6 (0110D)

1000 1001 **1111**BCD + 0110D = 1000 **1010** 0101BCD

1000 **1010** 0101BCD + 0110 0000D = 1001 0000 0101BCD

# Rozdíl v binární soustavě

23D – 17D = 6D

F3H – 2DH = C6H

## Rozdíl pomocí dvojkového doplňku v binární soustavě

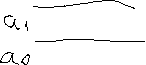
14D – 5D = 9D

1110B + 1011B = 11001

Dvojkový doplněk -5 0101 inverze 1010 + 1 = 1011

19. 9. 2023

# Kódování dat

1. Dle číselných soustav
   1. Binární
   2. Osmičkovou (Octálová)
   3. Šestnáctkovou (Hexadecimální)
   4. Desítkovou (Dekadickou)
2. BCD
   1. Každý znak kóduje pomocí 4 bitů binárního čísla
3. BCD+3
   1. 0d = 0011BCD+3
   2. 1D = 0100BCD+3
4. GRAYův kód
   1. Eliminuje hazardy
   2. U sousedních čísel mění 1 bit
      1. 0D = 0000G
      2. 1D = 0001G
      3. 2D = 0011G
      4. 3D = 0010G
      5. 4D = 0110G
5. Kontrolní kódy
   1. Kontrola paritou
   2. Hammingův kód
      1. Více paritních bitů u jednoho chybného bitu chybu lokalizuje
6. Čárový a QR kód
7. Magnetický kód
8. Ascii kód
   1. American Standart Code for Information a Interchange

2. 10. 2023

# Logické funkce a obvody

## Definice

* Logická funkce je funkce, která pro konečný počet vstupních parametrů vrací logické hodnoty
* Používá se v matematické logice, v oboru teorie řízení a číslicové techniky, v praxi pak například v mikroprocesorové technice. Parametry logické funkce jsou logické proměnné.

## Popis

* Přiřazuje-li logická funkce výstupní hodnoty všem kombinacím vstupních logických proměnných, pak se nazývá úplně zadaná logická funkce.
* V opačném případě se nazývá neúplně zadaná logická funkce.
* Kombinace vstupních logických proměnných, k níž není určena hodnota výstupní logické funkce, se nazývá neurčitý stav.

## Logické obvody

* Realizují logické funkce.
* Základní soubor: AND, OR a NOT
* Minimální soubor: NAND a NOR
* Další funkce: XOR, XNOR, implikace a inhibice

## AND (Konjuktor)

* Tento člen provádí funkci tzv. logického součinu (konjunkce).
* Y = A \* B

X1 X2 Y0 0 1  
0 1 0  
1 0 0  
1 1 1

## OR (Disjunktor)

* Tento člen provádí funkci tzv. Logického součtu (disjunkce).
* Y = A + B

X1 X2 Y  
0 0 0  
1 0 1  
0 1 1  
1 1 1

## NOT (Invertor)

* Y = (-A)

X1 Y  
0 1  
1 0

## NAND (Shefferova funkce)

* Tento člen provádí funkci tzv. Negovaného logického součtu (Shefferovu funkci) neboli součet negací schopno pracovat jako inventor. Lze pomocí něho realizovat většinu klopných obvodů.
* Y = -(A\*B)

X1 X2 Y  
0 0 1  
0 1 1  
1 0 1  
1 1 0

## NOR (Peircerova funkce)

* Tento člen provádí funkci tzv. Negovaného logického součtu (Peircerovu funkci)
* Y = -(A+B)

X1 X2 Y  
0 0 1  
0 1 0  
1 0 0  
1 1 0

11. 10. 2023

## XOR

* Tento logický člen vyčisluje exkluzivní logický součet
* Y = ((-A) \* B) + (A \* (-B))

X1 X2 Y  
0 0 0  
0 1 1  
1 0 1  
1 1 0

## XNOR

* Jedná se o negaci exkluzivního logického počtu
* Y = (A \* B) + ((-A)\*(-B))

X1 X2 Y  
0 0 1  
0 1 0  
1 0 0  
1 1 1

## Implikace

* Y = (-A) + B

X1 X2 Y  
0 0 1  
0 1 1  
1 0 0  
1 1 1

## Inhibice

* Y = (-A) \* B

X1 X2 Y  
0 0 0  
0 1 1  
1 0 0  
1 1 0

17. 10. 2023

# Integrované obvody (IO)

* Jedná se o spojení (integraci) mnoha jednoduchých elektrických součástek, které společně tvoří elektrický obvod vykonávající nějakou složitější funkci

## Dělení IO

* Monolitické
* Hybridní

## Monolitický IO

* Základem pro výrobu moderních monolitckých IO je monokrystal z velmi čistého polovodiče
* Hotový monokrystal(válcový či doutníkový tvar) se nařeže na tenké plátky (wafery).
* Na jednom plátku je několik řad a sloupců stejných obvodů
* Celý obvod je pak zapouzdřen do (většinou plastového) pouzdra.

## Hybridní IO

* Skládají se z keramické destičky
  + Na destičku se pomocí sítotisku nanesou vodivé spoje, rezistory a přilepeny křemíkové destičky s diskrétními polovodičovými součástkami nebo jednoduššími monolitickými integrovanými obvody (případně i kondenzátory či cívky).

## Další dělení (další kritéria)

* analogové nebo číslicové obvody
* stupeň integrace
  + SSI – malá integrace (Small Scale Integration)
  + MSI – střední integrace (Middle Scale Integration)
  + LSI – vysoká integrace (Large Scake Integration)
  + VLSI – velmi vysoká integrace (Very Large Scale Integration), někdy také XLSI (eXtra Large Scale Integration)
* unipolární a bipolární obvody
* programovatelné a neprogramovatelné obvody
* sériově a zakázkově vyráběné obovody

## SN 74 LS 00

* SN – výrobce
* 74 – teplota
* LS – vlastnost TTL (HC, HCT – vlastnost MOS)
* 00 – funkce

31. 10. 2023

## Zadání

* Vstupy
  + 2 bezpečnostní tlačítka pro ruce
  + nožní spouštěč lisu
* Výstupy
  + Lis
  + Alarm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stav | n | l | p | L | A |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 | 0 |

# Úplně normální funkce

* ÚNDF disjunktivní součet součinů (mintermů)  
  řádky kde výstup = 1  
  všechny vstupní proměnné  
  0 negovaný tvar 1 přímý tvar
* ÚNKF konjunktivní součin součtů (maxtermů)  
  řádky kde výstup = 0  
  všechny vstupní proměnné  
  1 negovaný tvat 0 přímý tvar

ÚNKF

A = (n + l + p) \* (n + l + -p) \* (n + -l + p) \* (n + -l + -p) \* (-n + -l + -p)

L = (n + l + p) \* (n + l + -p) \* (n + -l + p) \* (n + -l + -p) \* (-n + l + p) \* (-n + l + -p) \* (-n + -l + p)

8. 11. 2023

# Karnaughovy mapy (v bloku)

# Zadání 2

Napište tabulku pro funkci která generuje na výstupu jedničku když má měsíc 31 dní

celé v bloku

14. 11. 2023

# Minimalizace pomocí Karnaughovy mapy

Hodnoty x můžeme ale nemusíme použít

16. 11. 2023

# Postup návrhu kombinačního obvodu

1. Přepis slovního zadání do pravdivostní tabulky
2. Zápis Karnaughových map
3. Výpis funkcí (pro jedinou 1 nebo jedinou 0 lze psát ÚNDF nebo ÚNKF)
4. Kontrola správnosti návrhu simulací
5. Úprava výrazu pro nízký počet integrovaných obvodů
6. Kontrola správnosti návrhu simulací
7. Zapojení na kontaktním poli

## Zadání číslo ani bůh neví kolik

* Navrhněte logiku obvodu pro alarm, který se ozve, když je nesprávně, nebo vůbec nezadaný kód a v místnosti se otevře okno nebo dveře
* sicko mam v notysku :3

22. 11. 2023

**Kombinační obvody**

* Výstupy závislé pouze na vstupních kombinacích a ne na jejich předchozích hodnotách
* jediné kombinaci vstupních hodnot odpovídá jediná výstupní kombinace

**Sekvenční obvod**

* Hodnota výstupní veličiny závisí nejen na okamžité kombinaci hodnot vstupních veličin, ale i na posloupnosti hodnot vstupních veličin v předchozím čase – pamatuje si předchozí stav

# Kombinační xobvody

## Dělení kombinačních obvodů

* Kodéry, dekodéry, rekodéry
* Multiplexery, demultiplexery
* generátory parity
* digitální komparátory
* obvody pro aritmetické operace

## Dekodéry

* Mění jeden kód na druhý
  1. BCD na sedmisegment
     + 7446 a 7447 – výstup aktivni nule
     + 7448 a 7449 – výstup aktivní v logické 1
     + statický režim – každý displej má svůj dekodér
     + dynamický režim – pro všechny displeje je použit pouze jeden dekodér
  + sedmisegment se společnou anodou
    - KW1 402ASA, KW1 561ASA
  + 1 z N
    - vyhledejte IO pro DC 1 ze 4
    - 1 z 10 – 7442
    - 1 z 16 – 74154

## Multiplexery a demultiplexery

* Multiplexer (MPX, dataselektor)
  + elektronické přepínače logických signálů
  + přepínání je ovládáno výběrovým (adresovým) signálem
  + přenáší informaci z jednoho z N vstupů na jeden výstupní
* Další použití
  + přechod paralelní informace na sériovou (DMX opačně)
  + řešení log. Obvodů

20. 12. 2023

* Příklady IO pro 2MX
  + 74153 – 2 čtyřkanálové MX se společnou adresou
  + 74151 – osmikanálový MX s přímým a negovaným výstupem
* Demultiplexer (DMX)
  + přepíná v závislosti na výběrovém kódu jediný logický vstupní signál na jeden z N výstupů
  + DC lze použít pro funkci DMX

3. 1. 2024

## Generátory parity

* Slouží jako pomocný obvod u identifikace chyby v binárním slově
* Generuje jednoduchý výstup, tzv. Paritní bit, který je přenášen nebo uchován a použije se ke kontrole přeneseného (uchovaného) slova.
* Jeho výstupem je parita vstupního vektoru.
* **Na sudou**
  + pokud je celkový počet 1 **sudý**, tak do P pošle 1
* **Na lichou**
  + pokud je celkový počet 1 **lichý**, tak do P pošle 1
* 74180, 74280, 74286

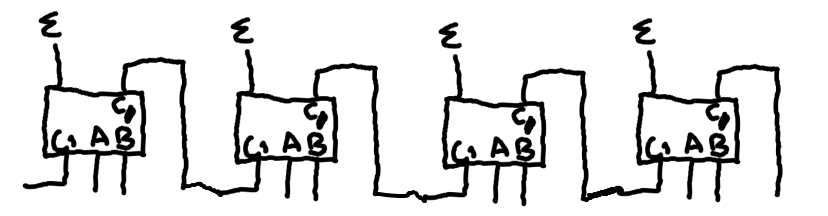
## Komparátory

* Kombinační logický obvod, který porovnává dvě binární čísla a generuje výstupní signál o jejich rovnosti, nebo různosti
* Dělíme je na:
  + Aritmetický – vyhodnocující >,=,<
  + Logický – vyhodnocující = a nerovná se

9. 1. 2024

* Příklad io 7485

## Sčítačky

1. **Sčítání jednobitového čísla**
   1. Cin = přenos z předchozího řádu
2. **Sčítání vícebitového čísla**
   1. **Sériové –** postupný součet vždy po 1.bitu
      * Používá posuvné registry
      * Pomalejší, jednodušší
      * Sčítá ve více taktech
   2. **Paralelní –** součet všech bitů datového slova najednou
      * Sčítá všechny bity na jednou
      * je to basically několik jednobitových sčítaček za sebou, každý bit = sčítačka (4 bitová paralelní = 4 jednobitové sčítačky)
      * Složitější, rychlejší
      * .74283
   3. **Sérioparalelní –** postupný součet po skupinách bitů

30. 1. 2024

# Sekvenční obvody

Co je paměť? Je to zpětná vazba realizována tím, že je výstup spojený se vstupem.

## Typy SLO

* RS
* D
* JK
* T

## Typy SLO dle připojení CLK

* Synchronní
* Asynchronní

## Typy SLO dle automatu

* Mealy
* Moore

**MEALY** Y = f(S;X)

**MOORE** Y = f(S)

# Klopné obvody

* Elektronické obvody
* S několika stabilními nebo nestabilními stavy
* Složené z hradel nebo jiných aktivních prvků
* Použití jako paměťový prvek nebo časovač
* Jsou to sekvenční logické obvody (mají paměť)

## Dělení dle stability

* **Astabilní klopný obvod (AKO)**
  + **0 stabilních stavů**
  + **např. Generátor impulsů**
* **Monostabilní klopný obvod (MKO)** 
  + **1 stabilní stav**
  + **např. Zpožďovací prvek**
* **Bistabilní klopný obvod (BKO)**
  + **2 stabilní stavy**
  + **např. Paměťový prvek**

## Dělení dle připojení CLK

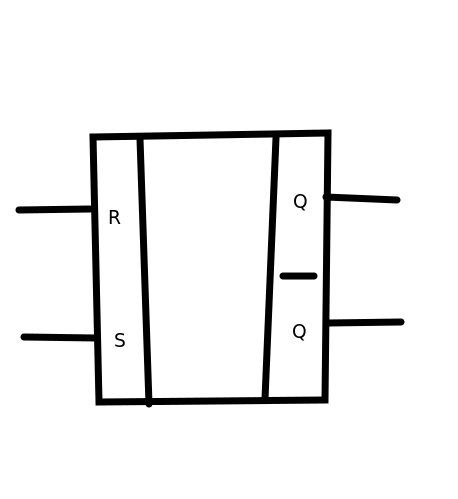
* Asynchronní
  + nemá vstup CLK
* Synchronní
  + má vstup CLK
* Statické řízení – úrovní 0 nebo 1
* Dynamické řízení – hranou

13. 2. 2024

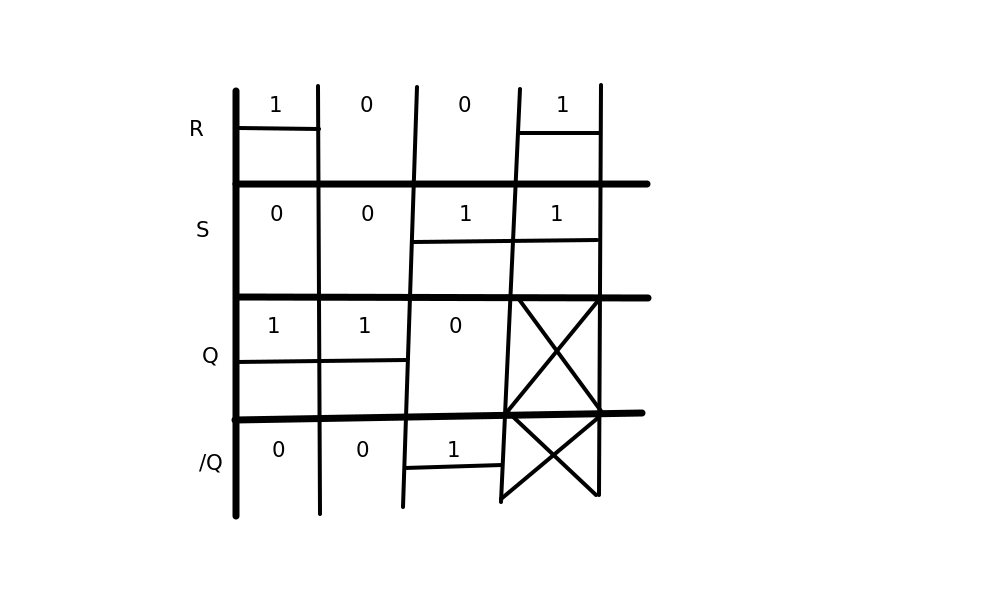
Stav je kombinace hodnot proměnných

# RS

* RS je jedním z nejjednodušších klopných obvodů.
* Vstup R označuje RESET – Q na logickou 0.
* Vstup S označuje SET – Q na logickou 1.

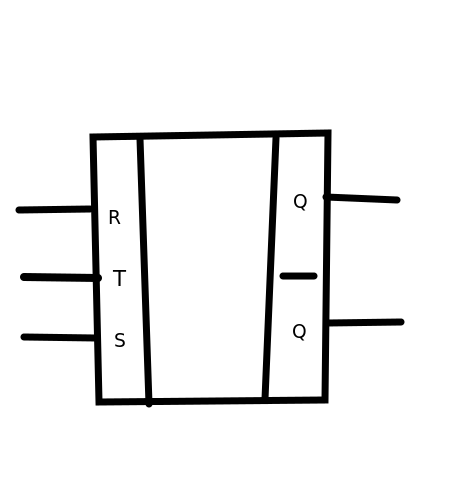


* R S Qt+1 t  
  1 0 0 0  
  0 1 1 1  
  0 0 Qt M  
  1 1 X X



# RST

* T – timer
* T = 0 => paměť
* T = 1 => pracuje jako RS klopný obvod



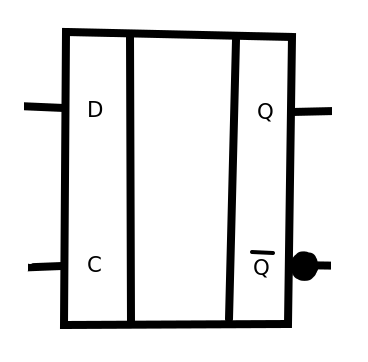
20. 2. 2024

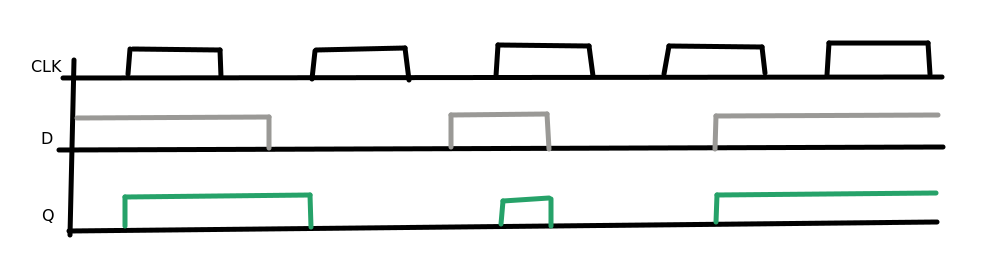
* **Nulová Q – 0 0**
* **Jednotková Q – 1 1**
* **Paměťová Q – Q M**
* **Klopná Q – /Q K**

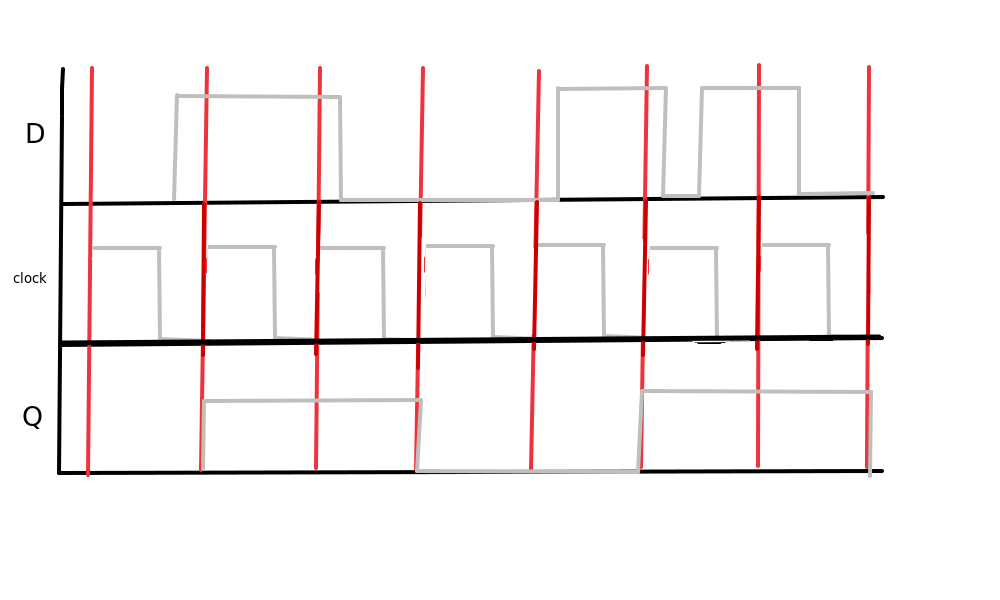
# D

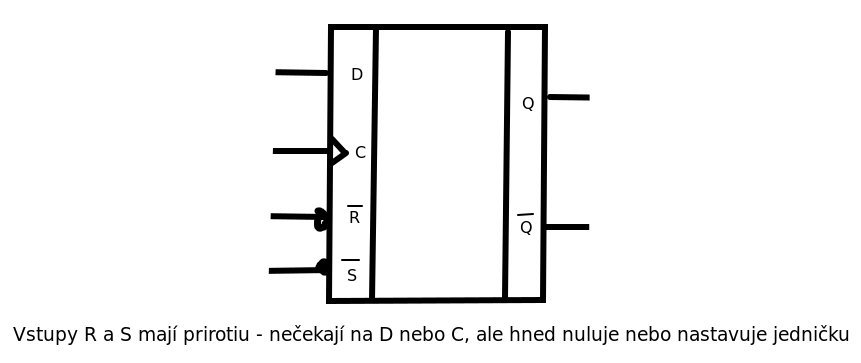
Obvod D realizuje jednobitovou paměť. Každý hodinový pulz způsobí zapamatování hodnoty vstupu.

Vyrábí se pouze jako synchronní obvod. Proto je jeho český název zdrž, anglicky delay, podrží tu hodnotu vstupu, která byla platná v době hodinového pulzu.

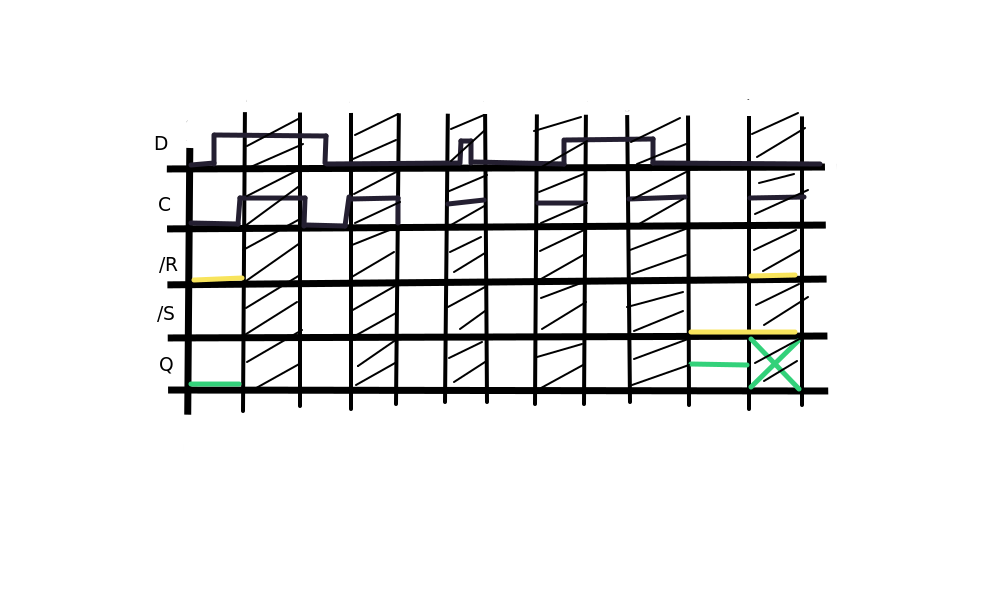




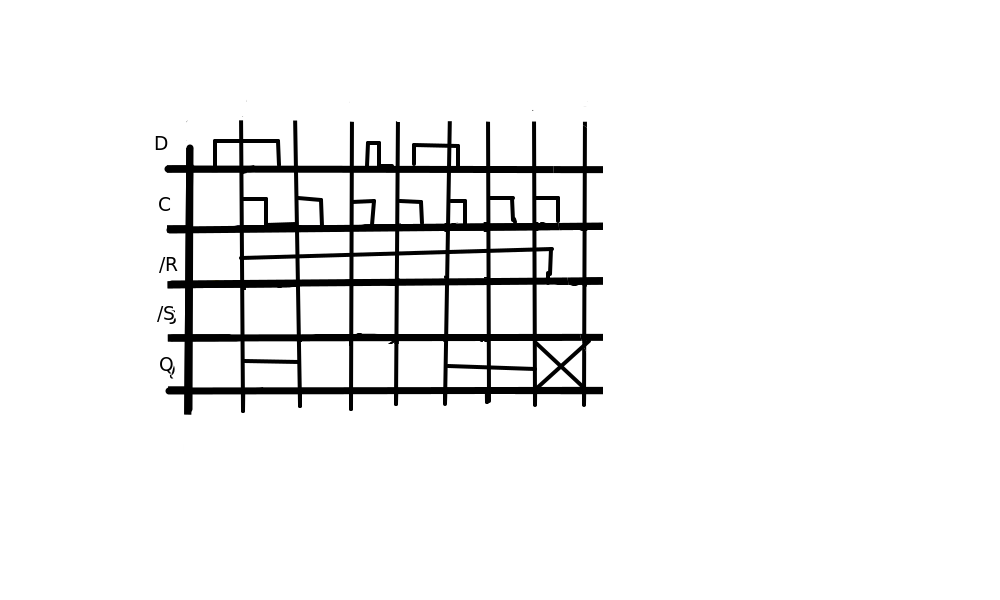


2024-02-27

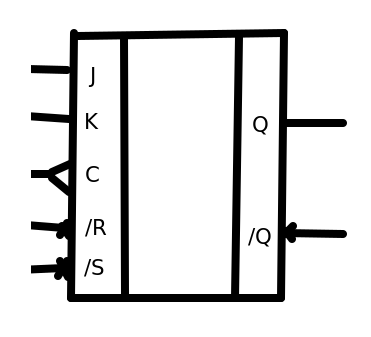
**dynamický stav**

****

**statický stav**

****

# JK



**J K Qt+1 .**

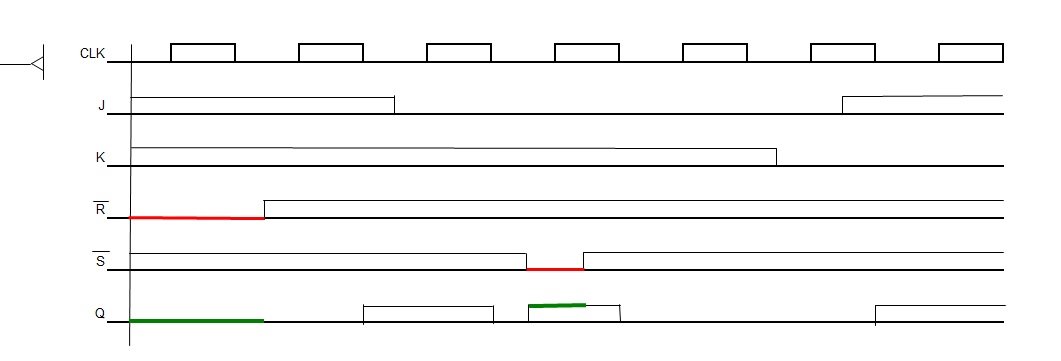
0 0 Qt M

1 0 1 1

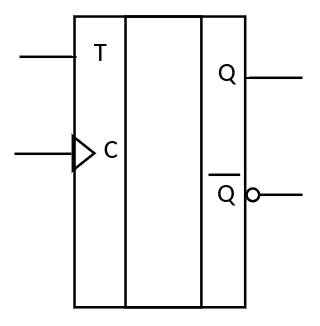
0 1 0 0

1 1 Qt K

pokud oba vstupy jsou 1, ale hodnota se překlopí do opačné



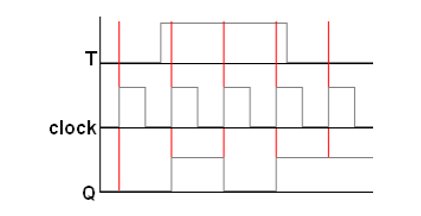
# T



**T Qt+1 .**

0 Qt M

1 /Qt K



26. 3. 2023

# Čítače

* Počítá impulzy přivedené na vstup, na výstupu je zobrazený příslušný kód.
* Modulo čítače = počet různých stavů

## Základní dělení

**Dle připojení CLK**

* Synchronní
* Asynchronní

**Dle směru čítání**

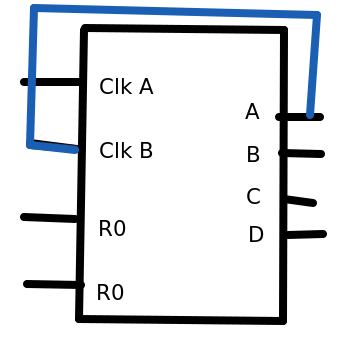
* Vpřed
* Vzad
* Obousměrné

**Dle kódu**

* Binární
* Dekadické
* Johnsonův
* A prakticky jakýkoliv jiný

**Dle využitých stavů čítače**

* S úplným cyklem
* Se zkráceným cyklem
* Integrované verze čítačů
  + 7490 – počítá 0 až 9
  + 7493 – počítá 0 až 15



* + 74193

## Asynchronní čítače

* S příchodem impulzu se překlopí pouze klopný obvod prvního řádu.
* Následné překlopení zařídí další stupně náběžnou nebo závěrnou hranou impulzů na vstupu násl. stupně (výstup předchůdce)

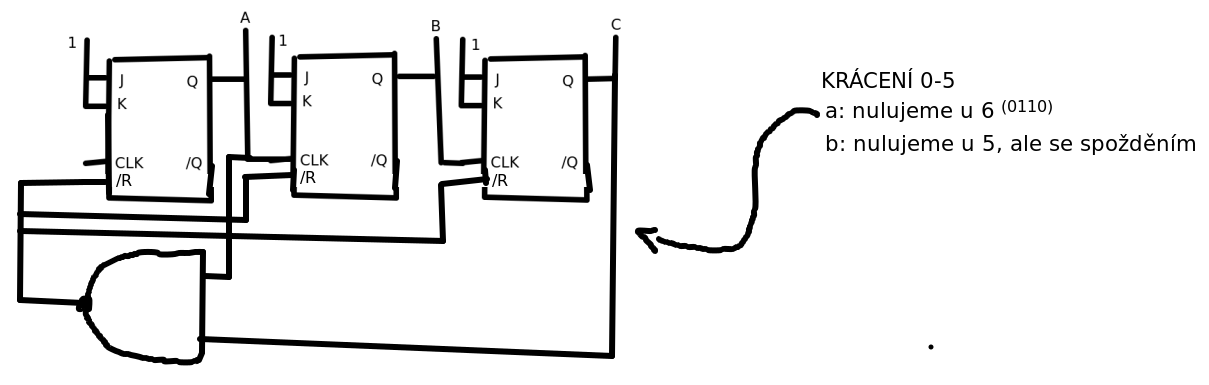
2024-04-09

* Synchronní čítače se většinou skládají ze dvou částí:
  + Paměťový element
    - implementován pomocí klopných obvodů
  + Kombinační element
    - bývají hradla, je však možná i realizace pomocí multiplexerů, paměti a pod

2024-04-30

## Krácení cyklu

Ke krácení cyklu budeme používat reset nebo set

2024-05-07

## Synchronní čítače

* Paměťový element
  + implementován pomoci klopných obvodů
* Kombinační element
  + bývají hradla, je však možná i realizace pomocí multiplexerů, pamětí a pod.

# Registry

* Používájí se k uchování nebo posuvu informace
* Registry a čítače jsou častým stavebním blokem v číslicových systémech.
* Statické registry jsou založeny na funkci synchronních klopných obvodů, nejčastěji KO typu D nebo KO typu JK.
* Rozdeleni podle práce s daty
  + datove
  + posuvne
  + kruhove
* Rozdeleni podle typu vstupu a vystupu
  + paraleln
  + seriove
  + seriove-paralelni
* Rozdeleni podle zpusobu zaznamenani informace
  + staticke
  + dynamicke

## Datový registr

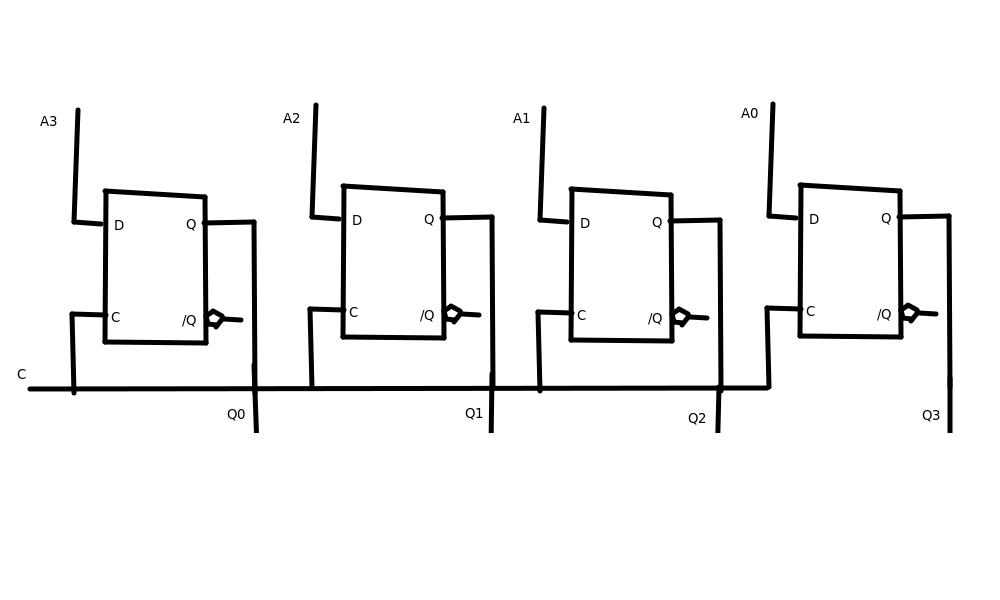
* Datový registr slouží k zachycení dat ve vhodných okamžicích a k jejich dočasnému pamatování.

## Posuvný registr

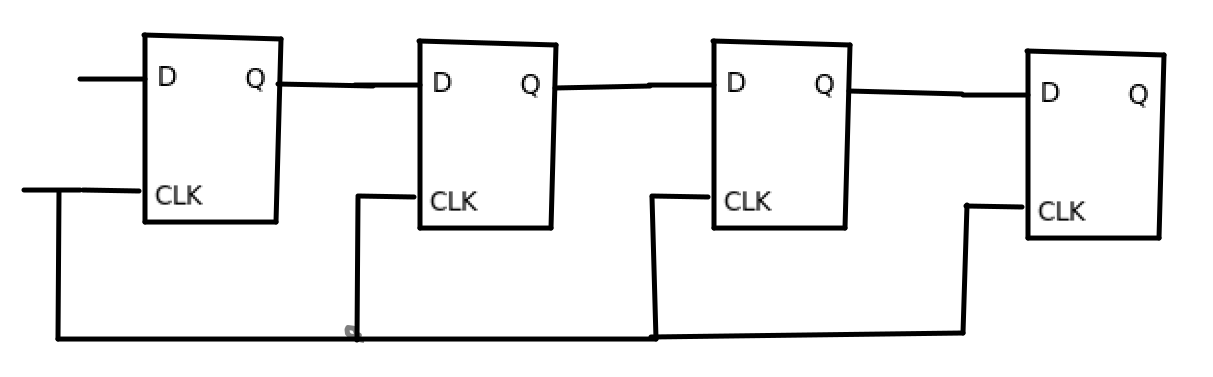
* Posuvný registr slouží k posouvání dat o jistý počet pozic doleva či doprava nebo kruhový registr, kde data rotují.

## Paralelní registr

* Skládá se z několika D KO spojenými hodinovým signálem
* Za jednoho C impulsu se celá informace zaznamená do registru až do dalšího impulzu

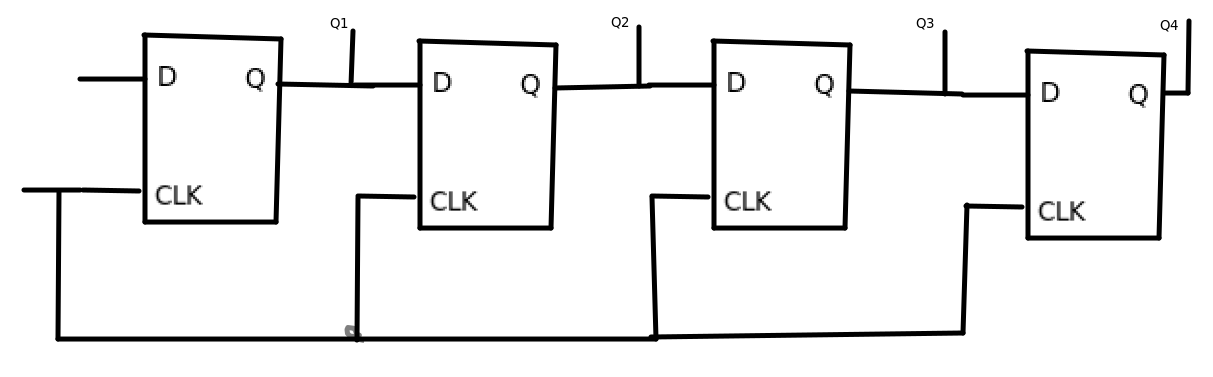


2024-05-14

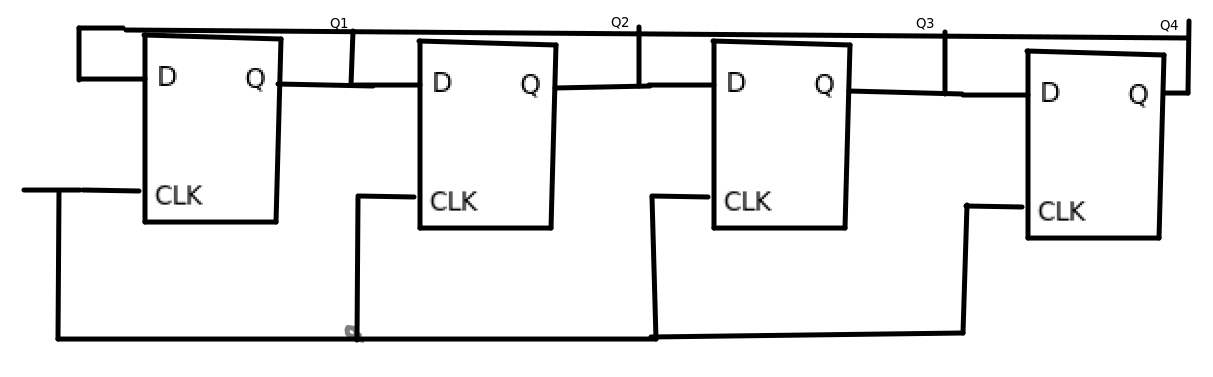


nad tímto je sériový registr

## Sériově-paralelní registr



Kruhovy



## Integrované verze

* 7495 – 4 bitový posuvný registr
  + s možností nahrání paralelních dat
* 7496 – 4 bitový posuvný registr
* 74194 – 4 bitový posuvný registr
  + nulování, nahrání paralelních dat, posuv vpřed, posuv vzad, paměť
* 74164 – 8 bitový posuvný registr
  + pouze sériové vstupy a paralelními výstupy

2024-05-27

# CCD Struktury

* Na začátku polovodičová struktura
* Ochuzená vrstva
* Na výstupu musí být tranzistor MOS
* Náboj v jámách lze ovládat fotoemisí
* Potenciálová jáma
* Velké napětí – inverzní vrstva

# Dynamicky posouvatelný

* Paměťová buňka
* S invertory MOS
* Náboj musí být obnovován
* Dva až čtyři hodinové impulzy

2024-06-04

# Programovatelné obvody

* Sekvenční a kombinační logický obvod
* Funkce obvodu není definována výrobou
* Před použitím je nutné jej naprogramovat

## Dělení

* Jednoduché programovatelné obvody
  + SPLD (PAL, GAL)
* Komplexní programovatelné obvody
  + CPLD
* Hradlová pole
  + FPGA

## PAL, GAL

* SPLD = Simple programable logic devices
* Umožňují naprogramovat pouze jednoduché logické obvody
* Mají předem definované vstupní a výstupní piny, uživatel pouze konfiguruje vlastní logickou funkci

## Další dělení SPLD

* PROM
* PAL
* PLA
* Na každém výstupním pinu je umístěna jedna jednobitová makrobuňka