



Elektronika a Mikroelektronika A4B34EM

10. přednáška

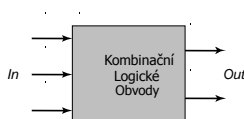
- Kombinační obvody
- Multiplexory
- Dekodéry, Kodéry
- Sekvenční obvody
- Klopné obvody
- Registry



Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL

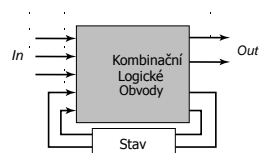


Kombinační vs. Sekvenční logika



Kombinační

$$\text{Výstup} = f(\text{In})$$



Sekvenční

$$\text{Výstup} = f(\text{In}, \text{Předchozí In})$$

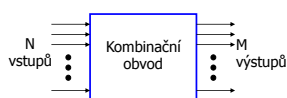


Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL



Kombinační logické obvody

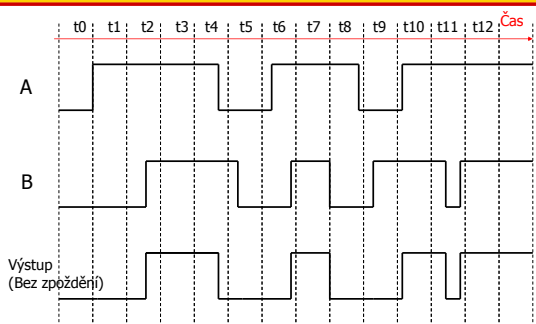
- Statická CMOS logika
 - Konvenční statická CMOS logika
 - Spínací Transistory/Přenosová hradla
- Dynamické CMOS logika
 - Domino logika



Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL



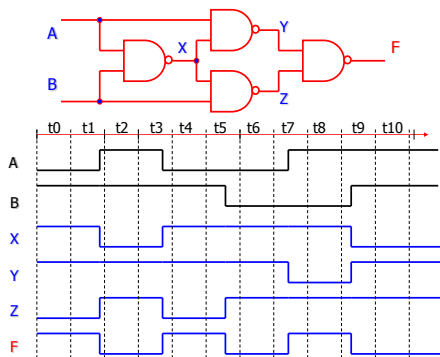
Časový diagram hradla AND



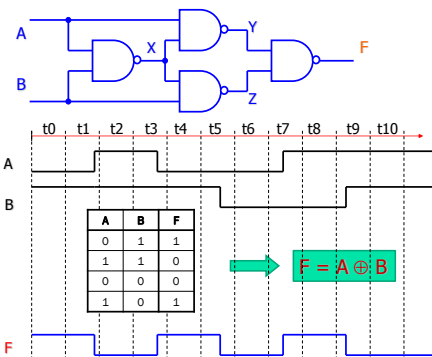
U kombinačních obvodů se výstup mění kdykoli s událostí na vstupech



Časový diagram příklad

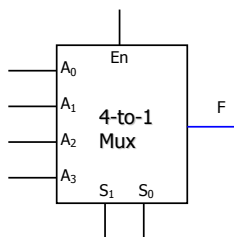


Časový diagram příklad





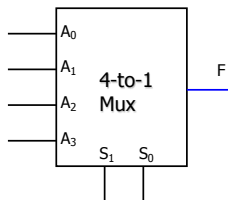
Multiplexory (Mux)



- Funkce: Výběr specifického vstupu
- Jeden z N vstupů (A) je přepojen na výstup F
- Výběrové bity (S)
- En (enable) bit může zakázat přenos na výstup F



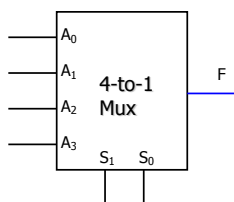
Multiplexor (Mux)



S1	S0	A3	A2	A1	A0	F
0	0	X	X	X	0	0
0	1	X	X	0	X	0
1	0	X	0	X	X	0
1	1	0	X	X	X	0
0	0	X	X	X	1	1
0	1	X	X	1	X	1
1	0	X	1	X	X	1
1	1	1	X	X	X	1



Multiplexor (Mux)



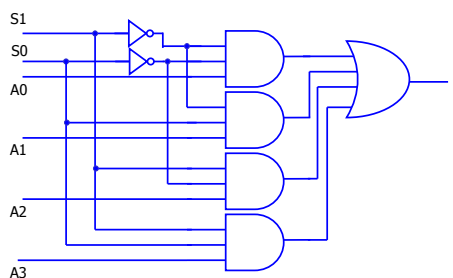
S1	S0	F
0	0	A0
0	1	A1
1	0	A2
1	1	A3

$$F = \overline{S_1}\overline{S_0}A_0 + \overline{S_1}S_0A_1 + S_1\overline{S_0}A_2 + S_1S_0A_3$$



Schema 4-to-1 Mux

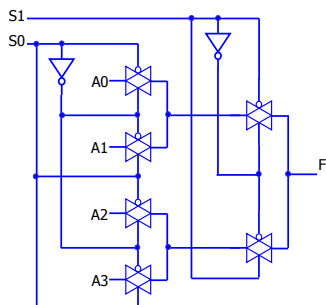
$$F = \overline{S_1}\overline{S_0}A_0 + \overline{S_1}S_0A_1 + S_1\overline{S_0}A_2 + S_1S_0A_3$$



Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL



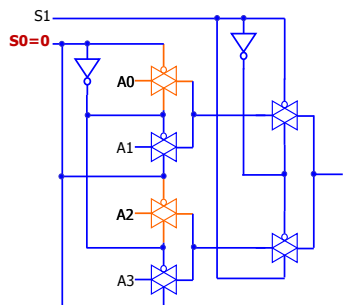
4-to-1 Mux pomocí CMOS přenosových hradel



S1	S0	F
0	0	A0
0	1	A1
1	0	A2
1	1	A3



4-to-1 Mux pomocí CMOS přenosových hradel

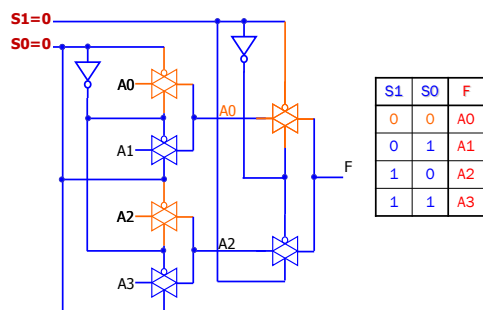


S1	S0	F
0	0	A0
0	1	A1
1	0	A2
1	1	A3

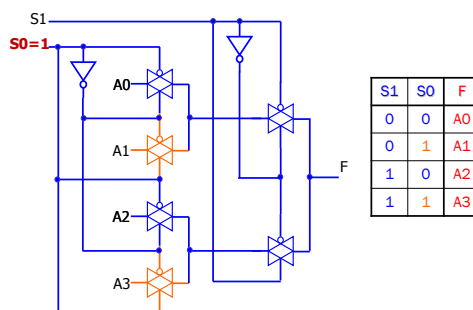




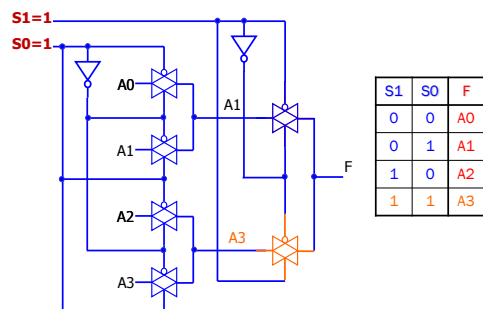
4-to-1 Mux pomocí CMOS přenosových hradel



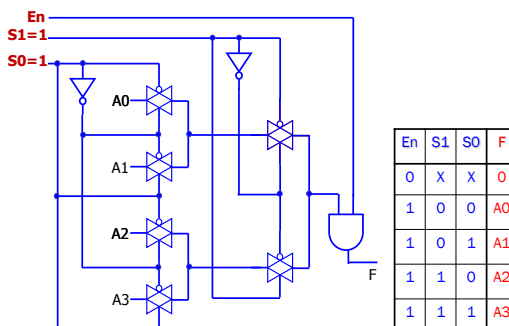
4-to-1 Mux pomocí CMOS přenosových hradel



4-to-1 Mux pomocí CMOS přenosových hradel



4-to-1 Mux pomocí CMOS přenosových hradel Signál Enable (F=0 když En=0)



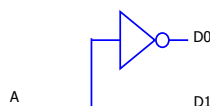
Dekodéry

- Dekodér je kombinační logický obvod, který ze vstupních dat v určitém kódu vytváří na výstupu jiná
- Má n adresových vstupů a 2^n výstupů, z nichž je jen jeden aktivní



1-to-2-Line Dekodér

- Binární dekodér či demultiplexor
- Obvod, který binárně kódovaný vstupní signál o n bitech převádí na 2^n výstupů, kódovaných jako 1 z 2^n



A	D1	D0
0	0	1
1	1	0

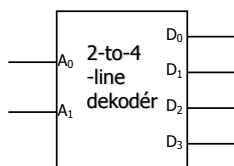
$$D_0 = \bar{A}$$

$$D_1 = A$$





N-to-M-Line Dekodér ($2^N \geq M$)



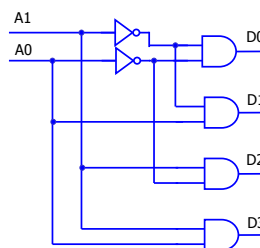
A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



μE



2-to-4-Line Dekodér



A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$D_0 = \overline{A_1} \overline{A_0}$$

$$D_1 = \overline{A_1} A_0$$

$$D_2 = A_1 \overline{A_0}$$

$$D_3 = A_1 A_0$$

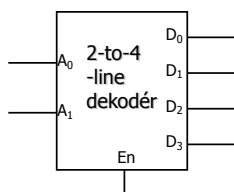
A jak to bude se signálem enable ?



μE



2-to-4-Line Dekodér se signálem Enable



En	A1	A0	D3	D2	D1	D0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$D_0 = \text{En} \overline{A_1} \overline{A_0}$$

$$D_1 = \text{En} \overline{A_1} A_0$$

$$D_2 = \text{En} A_1 \overline{A_0}$$

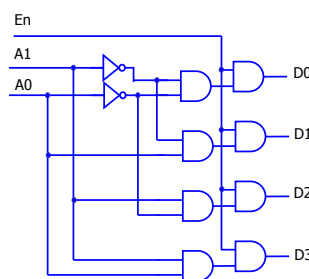
$$D_3 = \text{En} A_1 A_0$$



μE



2-to-4-Line Dekodér se signálem Enable



En	A1	A0	D3	D2	D1	D0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$D_0 = \text{En} \overline{A_1} \overline{A_0}$$

$$D_1 = \text{En} \overline{A_1} A_0$$

$$D_2 = \text{En} A_1 \overline{A_0}$$

$$D_3 = \text{En} A_1 A_0$$



μE



3-to-8-Line Dekodér

Pravdivostní tabulka

A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



μE



3-to-8-Line Dekodér

Pravdivostní tabulka

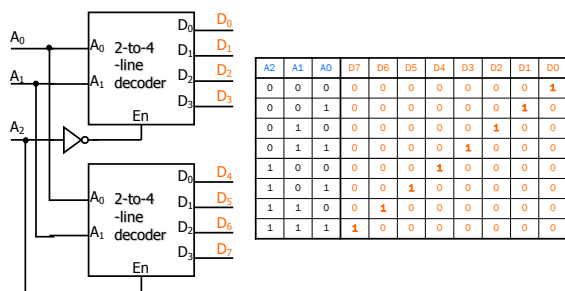
A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



μE

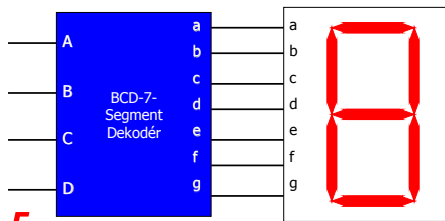


3-to-8-Line Dekodér



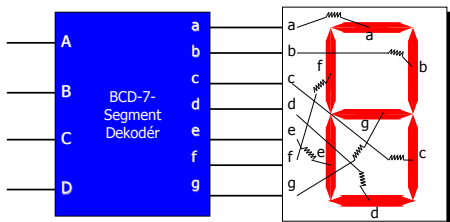
Dekodér BCD kódu na 7-Segmentový

- Převádí binárně kódovanou číslici na její zobrazení na sedmisegmentovém displeji
- BCD kód je definován z logické algebry binárního (dvojkového) kódu



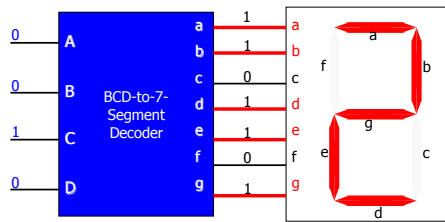
Dekodér BCD kódu na 7-Segmentový

- Pro každou vstupní číslici aktivuje příslušné segmenty



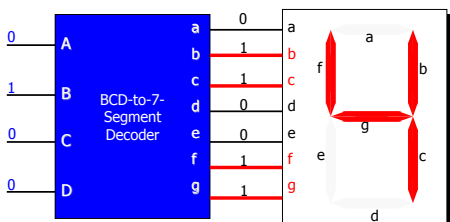
Dekodér BCD kódu na 7-Segmentový

- Dekóduje "2" a zobrazí 2



Dekodér BCD kódu na 7-Segmentový

- Dekóduje "4" a zobrazí 4



Dekodér BCD kódu na 7-Segmentový Pravdivostní tabulka

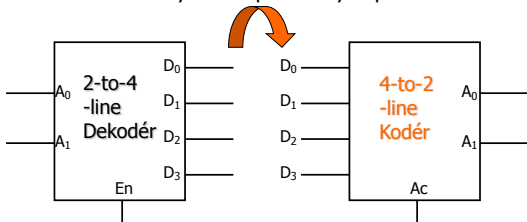
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	0	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	0	0	1	1	9
>10	Ostatní vstupy				0	0	0	0	0	0	0	





Kodér N z M ($M \leq 2^N$)

- Funkce dekodéru je inverzní k funkci kodéru.
- Jeho hlavním úkolem je redukovat počet vodičů v oblasti přenosu informace
- Má 2^n adresových vstupů a n výstupů



Kodér 3 ze 8

D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$D_x = 1$ pouze jedenkrát ve sloupci

$$A_0 = D_1 + D_3 + D_5 + D_7$$

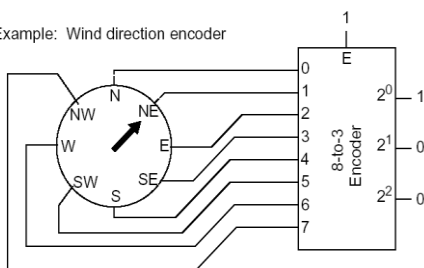
$$A_1 = D_2 + D_3 + D_6 + D_7$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$



Příklad Kodéru 3 z 8

Example: Wind direction encoder



Ukazuje pouze jediný směr v době čtení informace



Sčítačky

- kombinační logický obvod, realizující sčítání čísel, reprezentovaných v binární soustavě

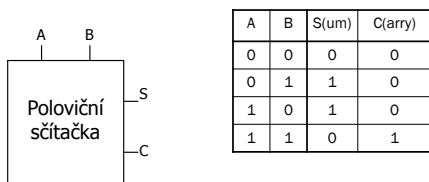
- **Poloviční sčítačka** (angl. Half adder) realizuje sčítání dvou jednomístných binárních čísel
- **Úplná sčítačka** (angl. Full adder) realizuje sčítání dvou jednomístných binárních čísel s připočítáním přenosu z předcházejícího řádu
- **N-bitová sčítačka s přenosem** (angl. Ripple carry adder, **RCA**) vznikne jednoduchým zřetězením N úplných 1-bitových sčítaček, a propojením výstupu sčítačky n -tého bitu se vstupem sčítačky $(n+1)$. bitu



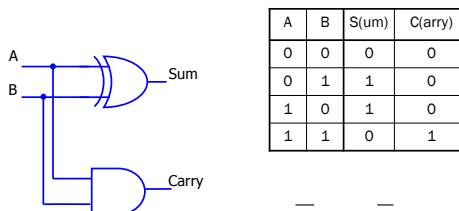
Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL



Poloviční sčítačka (1-bit)



Poloviční sčítačka (1-bit)



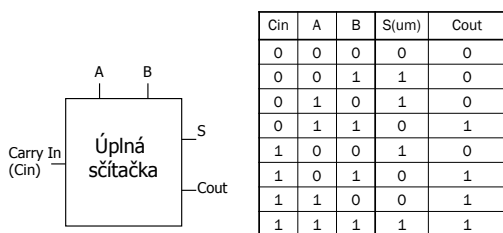
$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = AB$$

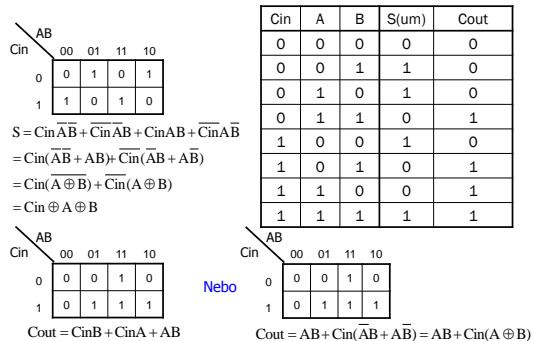




Úplná sčítačka



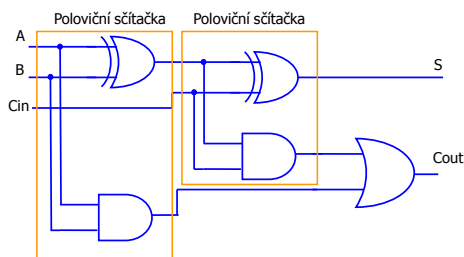
Úplná sčítačka



Úplná sčítačka

$$S = \text{Cin} \oplus A \oplus B$$

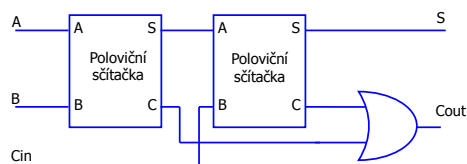
$$\text{Cout} = AB + \text{Cin}(A \oplus B)$$



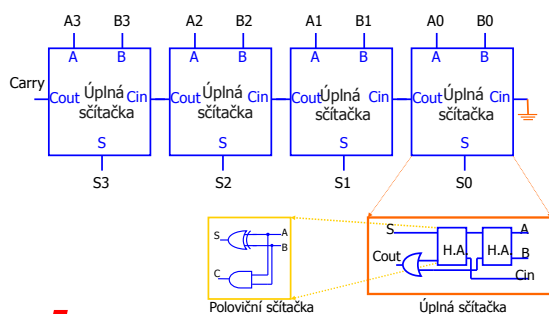
Úplná sčítačka

$$S = \text{Cin} \oplus A \oplus B$$

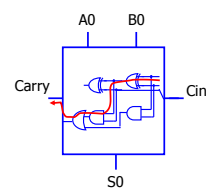
$$\text{Cout} = AB + \text{Cin}(A \oplus B)$$



4-bitová sčítačka s přenosem



Úplná sčítačka – zpoždění signálu

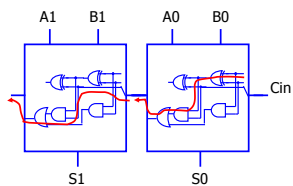


Kritická cesta
 $= 3 \text{ zpoždění hradel}$
 $= t_{\text{XOR}} + t_{\text{AND}} + t_{\text{OR}}$





Úplná sčítačka – zpoždění signálu

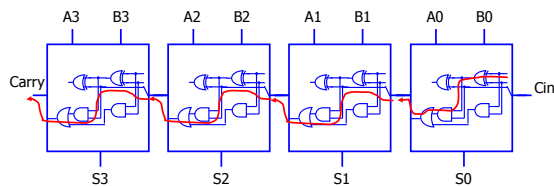


Kritická cesta 2. stupně
= 2 zpoždění hradel
= $t_{AND} + t_{OR}$
(protože 1. kritická cesta
> D_{XOR})

Kritická cesta
= 3 zpoždění hradel
= $t_{XOR} + t_{AND} + t_{OR}$



4-bitová Úplná sčítačka – zpoždění signálu

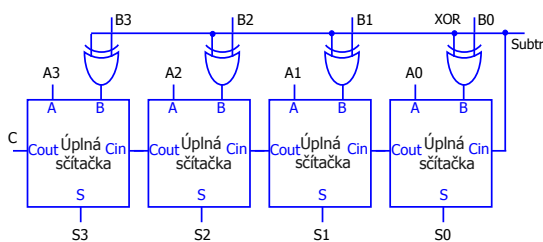


Kritická cesta = $t_{XOR} + 4 * (t_{AND} + t_{OR})$ pro 4-bitovou sčítačkou s přenosem (9 hradel)

Pro 4-bitovou sčítačkou s přenosem:
Kritická cesta $\sim 2(N-1)+3 = (2N+1)$ hradel



Bitová odčítačka



■ $A - B = A + (-B)$



Posuvný registr



Jiří Jakovenko – Elektronika a Mikroelektronika - Katedra mikroelektroniky – ČVUT FEL



Posuvné registry

- Směr posuvu
 - Doleva (násobení 2)
 - Doprava (dělení 2)
- Typy posuvu
 - Logické (unsigned)
 - Aritmetické (signed)



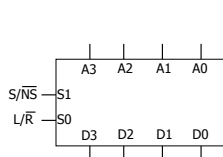
Posun logické hodnoty

- Registr posouvající vlevo
 - MSB: Odsunut pryč
 - LSB: nastaven na log. "0"
 - Příklady:
 - $(11001011 \ll 1) = 10010110$
 - $(11001011 \ll 3) = 01011000$
- Registr posouvající vpravo
 - MSB: nastaven na log. "0"
 - LSB: Odsunut pryč
 - Příklady:
 - $(11001011 \gg 1) = 01100101$
 - $(11001011 \gg 3) = 00011001$





4-bitový posuvný registr logické hodnoty



S1	S0	D3	D2	D1	D0
0	X	A3	A2	A1	A0
1	0	0	A3	A2	A1
1	1	A2	A1	A0	0

$$D_3 = \bar{S}_1 A_3 + S_1 S_0 A_2$$

$$D_2 = \bar{S}_1 A_2 + S_1 \bar{S}_0 A_3 + S_1 S_0 A_1$$

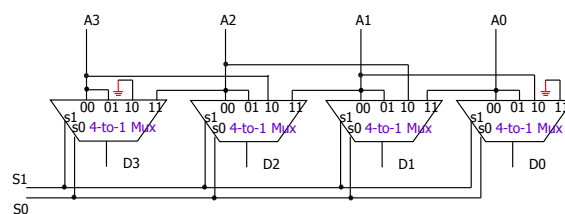
$$D_1 = \bar{S}_1 A_1 + S_1 \bar{S}_0 A_2 + S_1 S_0 A_0$$

$$D_0 = \bar{S}_1 A_0 + S_1 \bar{S}_0 A_1$$



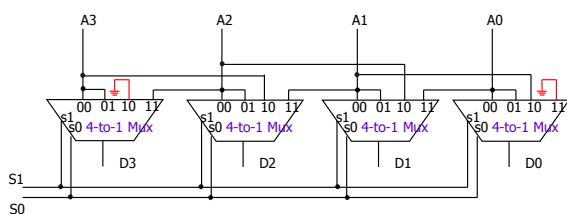
4-bitový posuvný registr logické hodnoty s 4-1 Multiplexorem

S1	S0	D3	D2	D1	D0
0	X	A3	A2	A1	A0
1	0	0	A3	A2	A1
1	1	A2	A1	A0	0



4-bitový posuvný registr logické hodnoty s 4-1 Multiplexorem

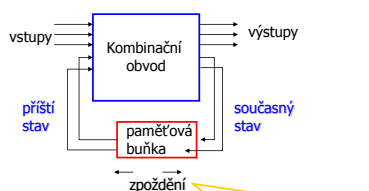
S1	S0	D3	D2	D1	D0
0	X	A3	A2	A1	A0
1	0	A3	A3	A2	A1
1	1	A2	A1	A0	0



Sekvenční obvody



Sekvenční logické obvody



■ Sekvenční obvody

■ Kombinační logický obvod

■ Stavová informace (uložená v paměťové buňce)

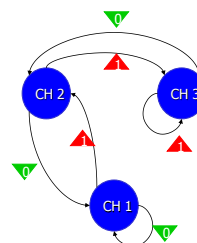
■ Výstup je funkcí vstupu a současného stavu

■ Můžou být synchronní a asynchronní



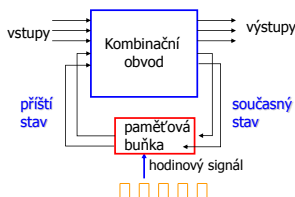
Příklad: stavový automat

TV dálkové ovládání





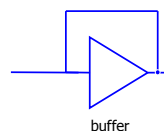
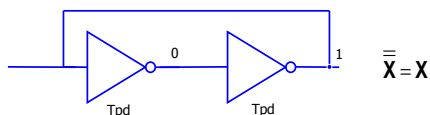
Sekvenční logické obvody



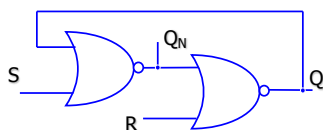
- Synchronní obvody mají hodinový signál k synchronizaci událostí
- Typický synchronní obvod má data uložená v paměti až do doby, kdy dojde k změně hodinového signálu



Uzavřená zpětnovazební smyčka – uchování logické hodnoty



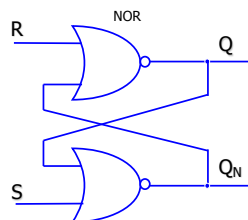
Klopný obvod RS



- RS je jedním z nejzákladnějších a nejjednodušších BKO
- Užívá se k zaznamenání přechodné informace
- Funguje jako elementární paměťová buňka



Klopný obvod RS



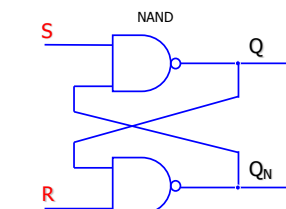
Pokud je na *R* a *S* zároveň logická 1, mluvíme o *zakázaném* nebo také *hazardním stavu*. Znamená to, že tento stav není definován a pokud nastane tato vstupní kombinace, není předem možné určit, v jakém stavu se bude nacházet výstup obvodu.

S	R	Q	QN
0	0	Q	Q
0	1	0	1
1	0	1	0
1	1	0	0

← Bez změny
← Reset
← Set
← Nedefinován



Klopný obvod RS

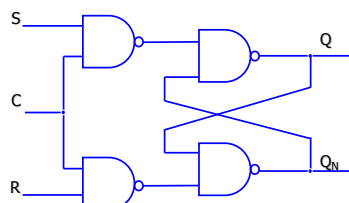


S	R	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	Q

← Bez změny
← Reset
← Set
← Bez změny



Klopný obvod RS řízený hodinovým signálem



Přidáním dalších dvou členů NAND zapojených jako blokování vstupů lze realizovat synchronní variantu tohoto klopného obvodu. Obvod tak bude reagovat na vstupy pouze s příchodem hodinového signálu C

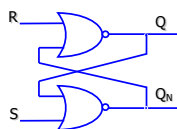
C	S	R	Q	QN
0	X	X	Q	Q
1	0	0	Q	Q
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

← Bez změny
← Bez změny
← Reset
← Set
← Nedefinován

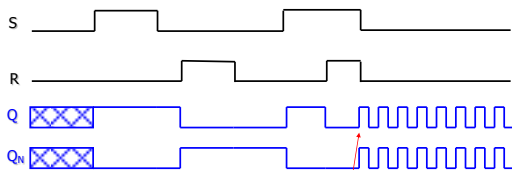




Stabilita klopného obvodu RS



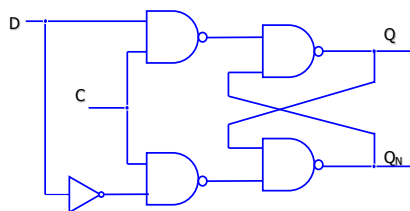
S	R	Q	QN
0	0	Q	Q̄
0	1	0	1
1	0	1	0
1	1	0	0



Nestabilní



Klopný obvod D – D Latch

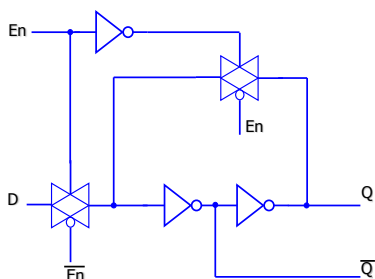


Realizuje jednobitovou paměť. Každý hodinový pulz způsobí zapamatování hodnoty vstupu.

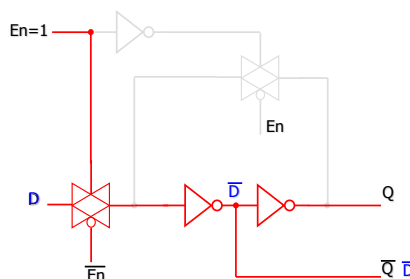
C	D	Q	QN
0	X	Q	Q̄
1	0	0	1
1	1	1	0



Klopný obvod D – D Latch z přenosových hradel



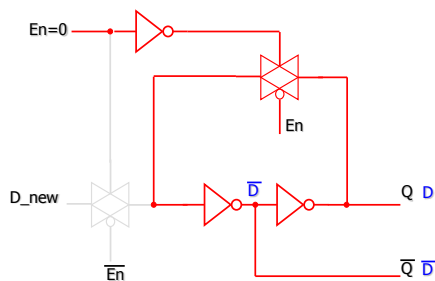
Klopný obvod D – D Latch z přenosových hradel



Zápis dat



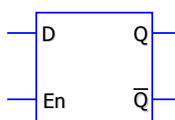
Klopný obvod D – D Latch z přenosových hradel



Pamatování dat



D schematická značka



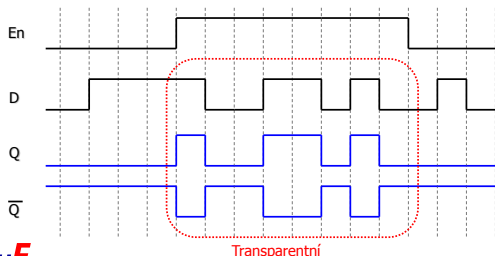
En	D	Q	Q̄
0	X	NC	NC
1	0	0	1
1	1	1	0



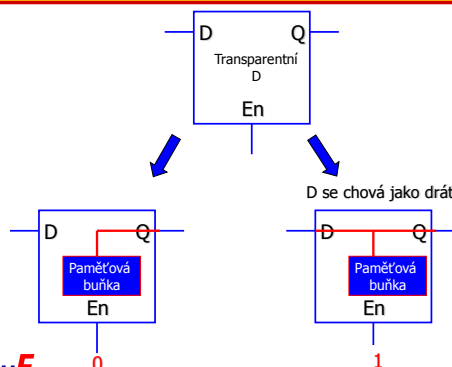


D Latch je transparentní

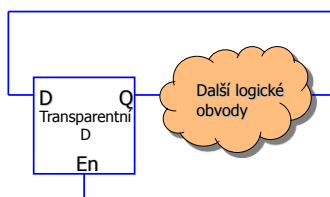
- D Latch je transparentní, výstup mění se změnou na vstupu
- Výstup stále sleduje vstup



Vlastnosti transparentního D



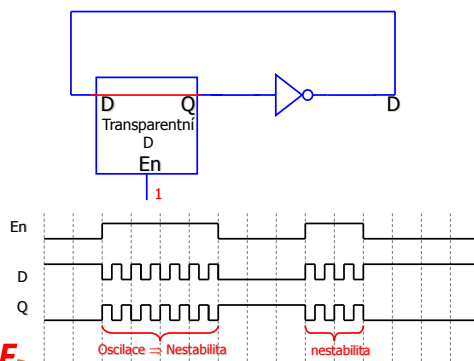
Problém transparentnosti



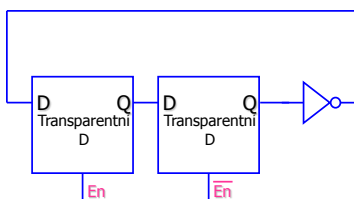
- Okamžitá hodnota na vstupu ovlivňuje další logické obvody
- Může vyvolat problémy se stabilitou a přenosem logické informace



Problém transparentnosti



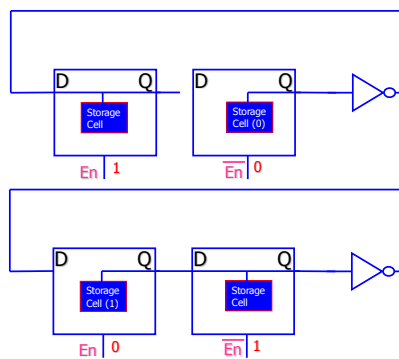
Odstranění transparentnosti



- Oddělení vstupu a výstupu, které jsou řízeny odděleně.
- Pouze jedno hradlo může přenést vstupní logickou hodnotu

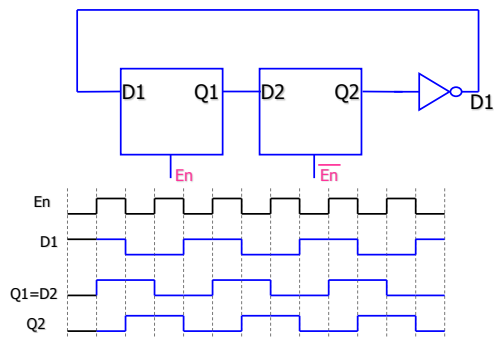


Klopný obvod D - Master-Slave

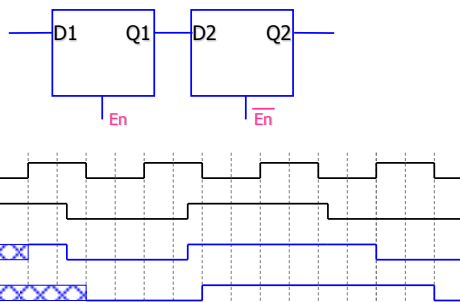




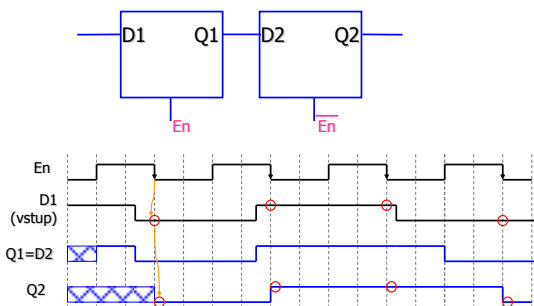
Klopný obvod D - Master-Slave



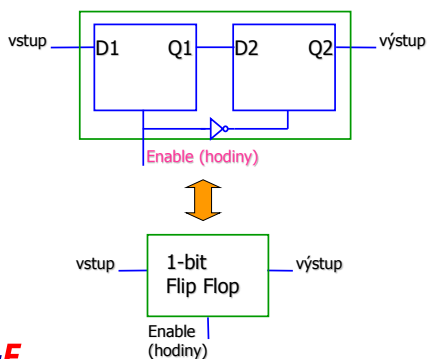
Klopný obvod D - Master-Slave



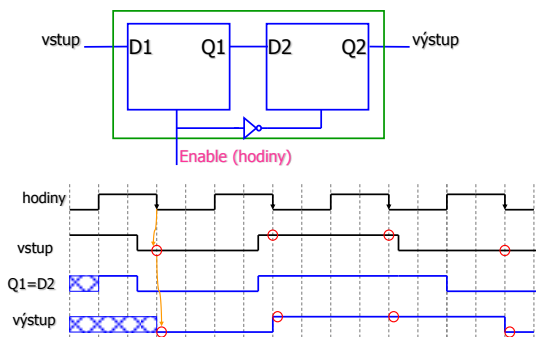
Klopný obvod D - Master-Slave



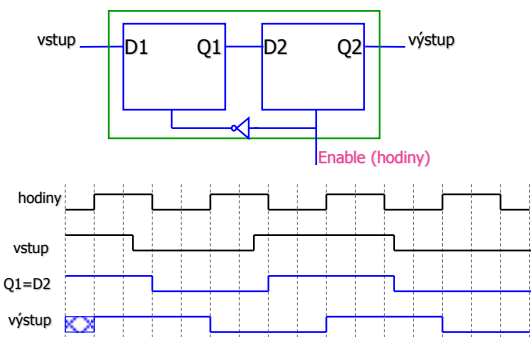
D Flip-Flop (F/F)



Flip-Flop řízený sestupnou hranou hodinového signálu

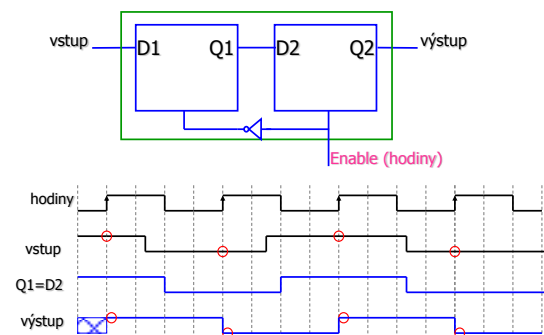


Flip-Flop řízený náběžnou hranou hodinového signálu

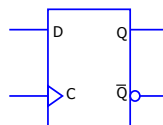




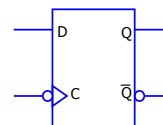
Flip-Flop řízený náběžnou hranou hodinového signálu



Flip Flops – schematické symboly



Flip-Flop řízený náběžnou hranou hodinového signálu

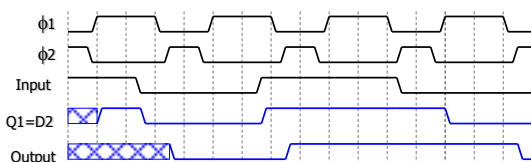


Flip-Flop řízený sestupnou hranou hodinového signálu

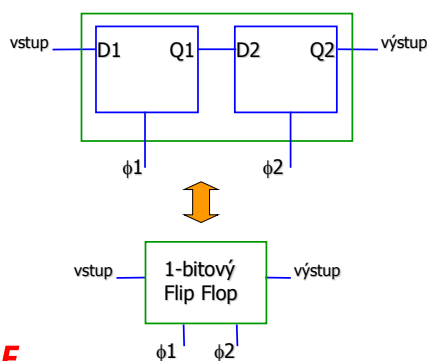


Dvoufázové hodiny

- V praxi řízení „enable“ není vhodné – vznik hazardu
- Bezpečné řešení pomocí tzv. dvoufázových hodin ($\phi 1$ a $\phi 2$)



Dvoufázové hodiny



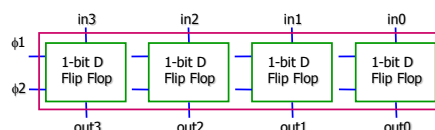
Registry

- Registr je malá paměť, do které lze uložit informaci o velikosti několika málo bitů.
- Velikost registru může být např. 8, 16, 32, resp. 64 bitů,...



4-bitový registr

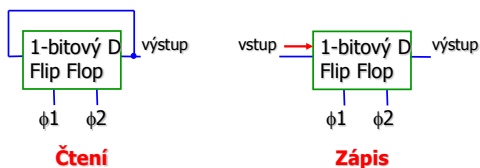
- Registr je nejzákladnější paměťová buňka,
 - každý mikroprocesor má mnoho registrů
 - např. x86 má 8 základních registrů
- Každý 1-bitový Flip-flop je jeden bit registru
- Spojení čtyřech 1-bitových Flip-flopů = 4-bitový registr



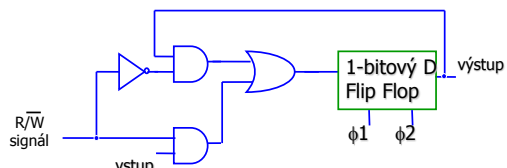


Řízení zápisu a čtení

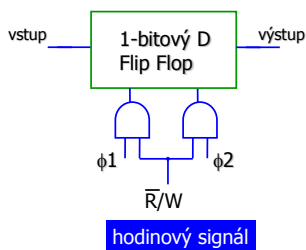
- Čtení: Získává data uložená v registru
- Zápis: zapíše data na vstupu
- $\phi 1$ a $\phi 2$ jsou dvoufázové hodiny



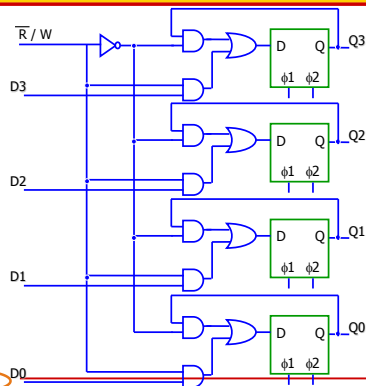
Řízení zápisu a čtení



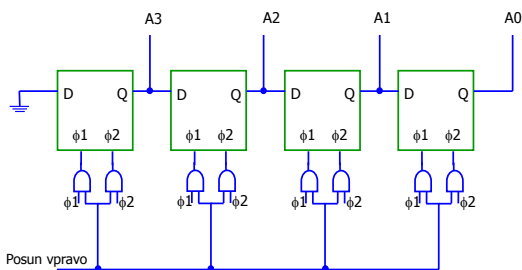
Jiná metoda Řízení zápisu a čtení



4-bitový Registr s paralelním zápisem



Posuvný registr logické hodnoty



Posuvný registr aritmetické hodnoty

