Otázka 10 - Y36SAP

Zadání

Logické obvody. Logické funkce, formy jejich popisu. Kombinační obvody a jejich návrh. Sekvenční systém jako konečný automat. Synchronní a asynchronní sekvenční obvody a jejich návrh. (Y36SAP)

Slovníček pojmů

literál - proměnná nebo její negace; reprezentuje logickou funkci

term - výraz tvořený pouze proměnnými (v přímém i negovaném tvaru) a operací logického součtu nebo logického součinu

P-term (součinový term) - term tvořený pouze proměnnými a operacemi log. součinu

S-term (součtový term) - term tvořený pouze proměnnými a operacemi log. součtu

minterm - takový P-term, který obsahuje všechny nezávisle proměnné

maxterm - takový S-term, který obsahuje všechny nezávisle proměnné

stavový index - desítkový zápis kombinace hodnot nezávisle proměnných

vstupní písmeno - kombinace hodnot vstupních proměnných

úplná normální disjunktivní forma (ÚNDF) - logický výraz tvořený součtem mintermů

úplná normální konjunktivní forma (ÚNKF) - logický výraz tvořený součinem maxtermů

mapa - čtverec nebo obdélník rozdělený na 2^n polí, kde n je počet nezávisle proměnných dané funkce

Logické obvody

Logický obvod je takový obvod, u něhož může každá veličina na vstupu i výstupu v ustáleném stavu nabývat s určenou přesností jen jedné ze dvou možných hodnot a který obsahuje takové prvky, jejichž vstupní a výstupní veličiny mohou nabývat také jen jedné ze dvou možných hodnot. Logický obvod je realizován skupinou logických členů vzájemně spojených tak, aby realizovali požadované logické funkce. Podle druhu realizované logické funkce dělimě logické obvody na:

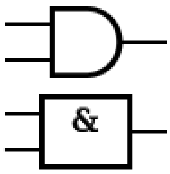
- kombinační logické obvody - jedná se o systémy, jejichž odezva je v určitém časovém okamžiku podmíněna výhradně hodnotami, které panují na vstupech tohoto systému
- sekvenční logické obvody - jedná se o systémy, jejichž odezva je v určitém časovém okamžiku dána nejen hodnotami proměnných na vstupech tohoto systému, ale i posloupností (sekvencí) předcházejících vstupních hodnot. Sekvenční obvod je proto opatřen pamětí, která svým stavem definuje vnitřní stav tohoto systému v závislosti na posloupnosti signálů, které přicházely na vstup. Formálně můžeme sekvenční logický obvod popsat dvěma soustavami rovnic, jednou pro výstupy a jednou pro vnitřní stavy.

Základní logické členy (hradla)

AND

Logický součin, značí se „&" nebo "·".

Značení



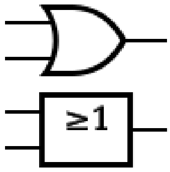
Pravdivostní tabulka

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

Logický součet, značí se "|" nebo "+".

Značení



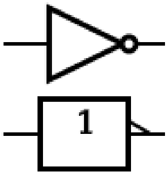
Pravdivostní tabulka

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT

Logická negace, značí se „~", "¬", nadtržítkem, někdy vykřičníkem.

Značení



Pravdivostní tabulka

X	Y
0	1
1	0

Logické funkce

Logická funkce je funkce, která pro konečný počet vstupních parametrů vrací logické hodnoty.

Používá se v oboru teorie řízení a číslicové techniky, v praxi pak například v mikroprocesorové technice. Parametry logické funkce jsou logické proměnné.

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.

Pro n logických proměnných lze definovat 2^{2^n} **logických funkcí.**

Booleova algrebra

Booleova algrebra je matematická disciplína, která je přímo aplikovatelná při návrhu číslicových obvodů. Zahrnuje pravidla a teorémy pro operace s logickými proměnnými a funkcemi. Při používání pavidel se využívají tři základní operace

- **logický součin (konjunkce)**
- **logický součet (disjunkce)**
- **negace (inverze)**

které tvoří teoretický prostředek pro návrh (syntézu) logických obvodů s požadovaným chováním. Vztahy mezi dvouhodnotovými proměnnými lze definovat několika matematickými zákony, tzv. *zákony Booleovy algebry*.

komutativní zákony x + y = y + x, x y = y x

asociativní zákony (x + y) + z = x + (y + z), (x y) z = x (y z)

distributivní zákony (x + y) z = x z + y z, x y + z = (x + z) (y + z)

zákon o vyloučeném třetím x + x' = 1, x x' = 0

zákon o neutrálnosti nuly x + 0 = x

zákon o neutrálnosti jedničky x·1 = x

zákon agresivity nuly x · 0 = 0

zákon agresivity jedničky x + 1 = 1

zákon o idempotenci prvků x + x = x, x · x = x

zákon absorpce x + x · y = x

zákon absorpce negace x + x' y = x + y, x (x' + y) = x y

zákon dvojité negace (x')' = x

De Morganovy zákony x' y' = (x + y)', x' + y' = (x y)'

Formy popisu logických funkcí

- Pravdivostní tabulka
- Logický výraz
- Vénnův diagram
- Zobrazení pomocí map
- Zobrazení na n - rozměrném tělese (nebudu popisovat)

Pravdivostní tabulka

Pravdivostní tabulka je nejběžnějším způsobem popisu logické funkce. popisuje zcela přesně chování logického obvodu, ale neobsahuje žádný návod pro jeho realizaci. Můžeme tedy na ni pohlížet jako na model chování logického systému. Obsahuje výčet všech kombinací vstupních proměnných a jim odpovídajících výstupů. Má-li logická fukce *n* nezávislých proměnných, bude mít pravdivostní tabulka 2^n řádků.

Ukázka prdvidostní tabulky sčítačky:

Operandy			Výstupní funkce	
A_n	B_n	C_n	C_{n+1}	S_n
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Pravdivostní tabulkou můžeme vyjádřit *určitou* i *neurčitou* funkci. Příkladem neurčité logické funkce je funkce f_2 z příkladu. Ve dvou jejích řádcích je zapsán symbol x, který značí, že při těchto kombinacích vstupních proměnných je lhostejno, jeslti logická funkce bude mít hodnotu 1 nebo 0. Z pravdivostní tabulky můžeme získat *logický výraz* pro jednotlivé logické funkce. Logický výraz funkce f_1 může být z pravdivostní tabulky získán dvěma způsoby:

- součtovou formou
- součinovou formou

podle toho, jestli použijeme k popisu logické funkce řádky, v nichž je funkce jedničková, nebo rádky, v nichž je nulová.

- *Základní součinový člen* je součin, který obsahuje všechny vstupní proměnné. Např. při vstupních proměnných a,b,c může mít základní součinový člen tvar a·¬b·c . Základní součinový člen se označuje pojmem *minterm*.

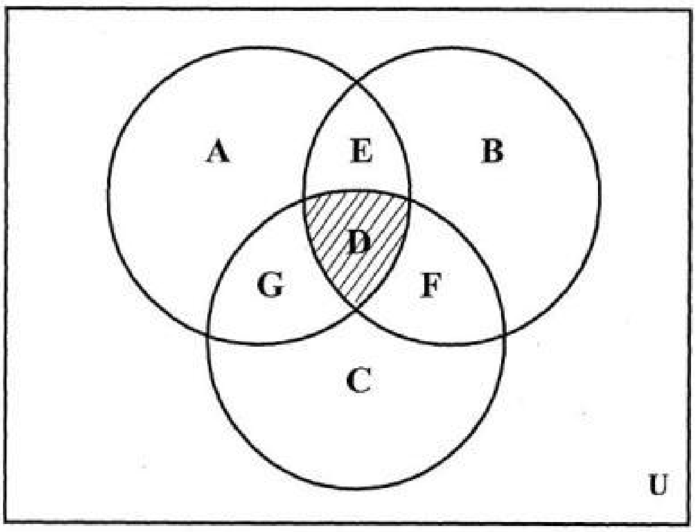
- *Základní součtový člen* je součet, který obsahuje všechny vstupní proměnné. Při vstupních proměnných a,b,c může mít základní součtový člen např. tvar ¬a+b+¬c . Základní součinový člen se označuje pojmem *maxterm*.
- *Úplná součtová forma* logické funkce je dána součtem základních součinových členů (mintermů), ve kterých logická funkce nabývá hodnoty 1.
- *Úplná součinová forma* logické funkce je dána součinem základních součtových členů (maxtermů), ve kterých logická funkce nabývá hodnoty 0.

Logický výraz

Logický výraz je popisem logické funkce pomocí logických (Booleovských) proměnných ve formě analytického popisu. Jeho vyjádření získáme zápisem logické funkce pro jednotlivé vstupní kombinace v součtové nebo součinové formě. Každému logickému výrazu odpovídá jednoznačně obvodová struktura za předpokladu, že máme k dispozici všechny realizační členy odpovídající loperátorům výrazu. Logický výraz tedy můžeme považovat za model struktury logického obvodu.

Vénnův diagram

Vénnův diagram je názorným množinovým způsobem zobrazení logické funkce. V rovinné oblasti zvolíme tolik dílčích podoblastí, kolik máme vstupních proměnných. Podoblasti volíme tak, aby existoval neprázdný průnik kterékoli možné kombinace těchto podoblastí. Jednoznačné přiřazení nastane tehdy, dohodneme-li se, že vstupní proměnná bude jedničková uvnitř příslušné podoblasti.

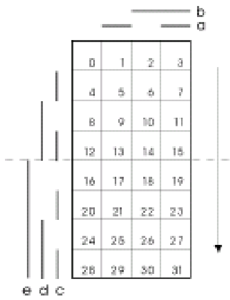


Zobrazení pomocí map

Zobrazení logické funkce je vedle pravdivostní tabulky nepoužívanějším způsobem zobrazení logických funkcí. Použítí map je výhodné, pokud počet nezávislých proměnných nepřekračuje čtyři a při prostorové interpretaci maximálně šest.

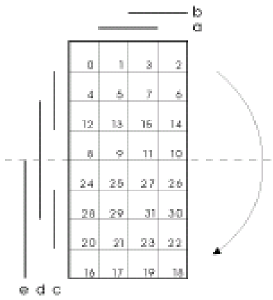
Mapa je v podstatě transformací pravdivostní tabulky, kdy každému řádku tabulky odpovídá jednoznačně jedno pole mapy. Každému řádku, resp. sloupci mapy přiřadíme jednu z kombinací nezávisle proměnných a provedeme zakódování řádků a sloupců. V každém poli mapy je zapsána hodnota logické funkce, která odpovídá logickým proměnným příslušného rádku a sloupce.

Svobodova mapa využívá účelu kódování řádek a sloupců přímý binární kód. Stavové indexy rostou



zleva doprava po sloupcích a shora doů po proměnných.

Karnaughova **mapa využívá pro kódování Grayův kód, tj. kód se změnou v jednom řádu. Sousední políčka v této mapě jsou sousední i ve smyslu binární souslednosti, kdy se dvě vstupní písmena liší pouze v hodnotě jedné vstupní proměnné. Tato mapa je výhodnější pro zjednodušování logických funkcí a pracuje se s ní snadněji. při vetším poču vstupních proměnnách je však méně přehledná než mapa**



Svobodova.

Kombinační logické obvody

Jedná se o takové obvody, u nichž mohou vstupní i výstupní proměnné v ustáleném stavu nabývat jedné ze dvou možných hodnot, logické nuly nebo logické jedničky. Kombinační logický obvod je realizován spojením základních logických členů tak, aby splňoval pozadovanou logickou funkci. Okamžitá hodnota výstupních proměnných kombinačního logického obovodu je dána pouze *okamžitou* kombinací vstupních proměnných. To znamená, že těmito obvody realizujeme výhradně takové situace, které nejsou závislé na předchozích kombínacích vstupů. Typickými představiteli kombinačních logických obvodů jsou

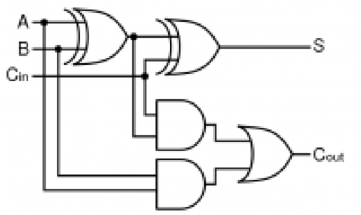
- **dekodéry**
- **multiplexery a demultiplexery**
- **komparátory**
- **obvody pro aritmetické operace**
-

Návrh logických obvodů

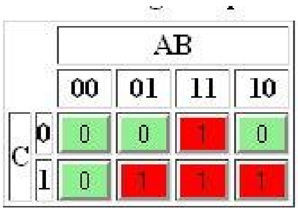
Postup návrhu kombinačního logického obvodu můžeme shrnout do následujících bodů:

- **slovní zadání logické funkce**
- **popis logické funkce obvodu (obvykle pomocí pravdivostní tabulky)**
- **minimalizace logické funkce**
- **kontrola správnosti navržené logické funkce**
- **realizace kombinačního lohického obvodu pomocí logických členů**

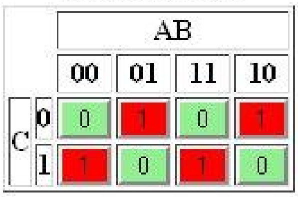
Sčítačka - plná



U sčítačky si ukážeme konkrétní příklad návrhu kombinačního obvodu. Na obrázku vidíme, jak taková sčítačka vypadá. Kromě vstupů a a b má ještě vstup p (ten slouží pro případ přenosu - pokud v minulé dvojici sčítaných bitů nastalo 1+1=10, tak se p nastaví na 1). S je tříbitový XOR a,b,p. q signalizuje již zmiňovaný přenos(může se definovat i jako majorita (víc jedniček nebo nul) z a,b,p).



Karnaugh Map



$F(ABC)= A B + B C + A C$

$F(ABC)= \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,B\,C + A\,\overline{B}\,\overline{C}$

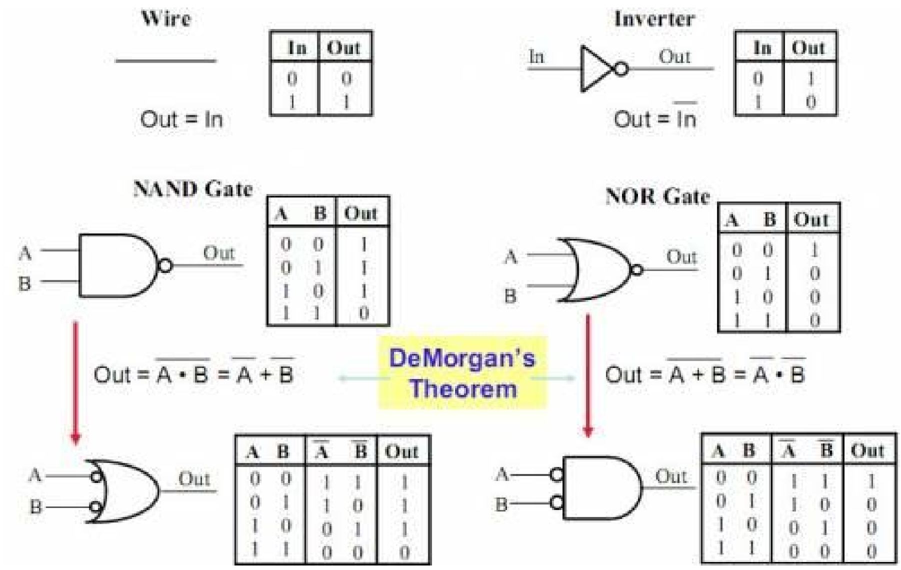
Vytvoříme si pravdivostní tabulku- q je tedy jedna pokud jsou aspoň dvě hodnoty z a,b,p jedničkové a s je jedna pouze při lichém počtu jedniček (při sudém nastane zmiňovaný přenos 1+1=0). Nyní přepíšeme hodnoty do Karnaughovy mapy.

a	b	p	q	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

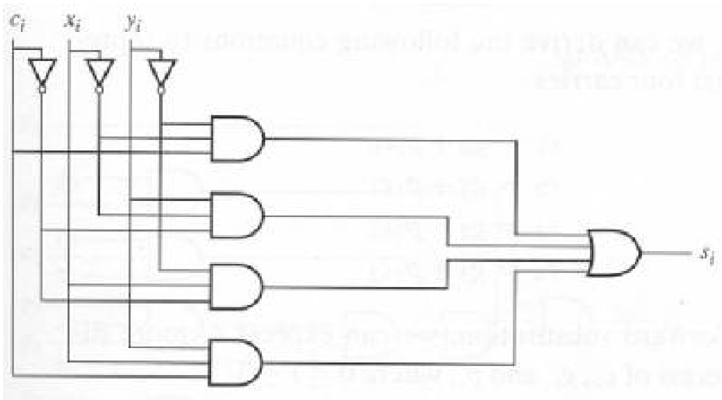
$s = \overline{a}\,\overline{b}p + \overline{a}b\overline{p} + a\overline{b}\,\overline{p} + abp$

$q = \overline{a}bp + a\overline{b}p + ab\overline{p} + abp$

Pro případ výsledku S se ukázal, že nejde dále minimalizovat. Pro přenos vyšla ÚNDF AB+BC+AC. Nyní již můžeme navrhnout obvod pomocí hradel. Na obrázku pro připomenutí vidíme invertovaná hradla



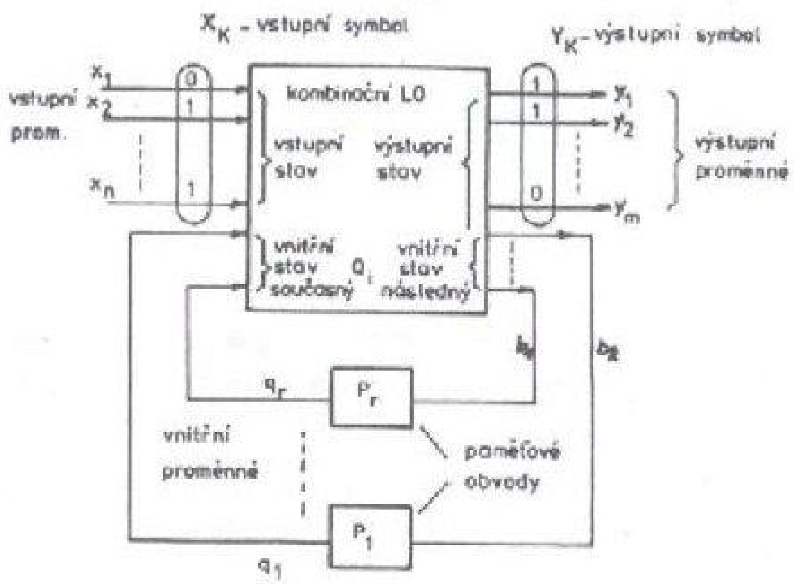
Pro případ S bude výsledný obvod vypadat takto: (c odpovídá našemu p)



Část obvodu s přenosem bude vypadat podobně, jen tam již nebudou invertory a vystačíme si se třemi hradly AND (podle rovnice S=AB+BC+AC).

Sekvenční systém jako konečný automat

Obecný model je zobrazen na následujícím obrázku



Obecná struktura SLO vypadá tak, že vstupem obvodu jsou mimo řádných vstupů i zakódované vnitřní stavy, které jsou přes paměťový člen (klopný obvod) přivedeny z výstupů obvodu (mimo řádných výstupů).

SLO formálně popisujeme konečným automatem, což je uspořádaná šestice A = <X, Y, Q, δ, λ, Q0>, kde jednotlivé symboly mají následující význam:

- **X - vstupní abeceda (konečná množina všech vstupních pismen, resp. symbolů)**
- **Y - výstupní abeceda (konečná množina všech výstupních písmen, resp. symbolů)**
- **Q - množina stavů (konečná množina všech vnitřních stavů)**
- **δ - stavově přechodová funkce (tj. zobrazení δ: X * Q → Q)**
- **λ - výstupní funkce (tj. zobrazení lambda: X*Q→Y nebo λ: Q → Y)**
- **Q0 - počáteční stav (Q0 je z Q)**

Rozlišujeme tyto základní typy automatů:

- **Autonomní automat – systém nemá vstupní proměnné – např. synchronní generátor kódu**
- **Kombinační automat – systém nemá vnitřní stavy – běžný kombinační obvod (nemá paměťové členy)**

- Mealyho automat – nejobecnější systém – výstup závisí jak na vstupech, tak na okamžitém vnitřním stavu automatu
- Mooreův automat – výstupní symbol závisí jen na vnitřním stavu automatu, vstupní symboly přepínají vnitřní stavy

Automaty popisujeme jejich tabulkou přechodů (nový stav pro každou kombinaci současného stavu a vstupu) a tabulkou výstupů (výstupní hodnota buď pro každou kombinaci současného stavu a vstupu – Mealy, nebo výstupní hodnota pro každý současný stav – Moore – na vstupu u něj výstup nezávisí).

Automaty typu Moore a Mealy

Moore

výstup závisí pouze na vnitřním stavu. Tabulka přechodů a výstupu je jednoduší protože výstupy jsou určeny jednoznačně pro každý vnitřní stav automatu a nezávisí na vstupním X. Když jej kreslíme, výstup kreslíme do koleček (uzlu) a na přechodných hranách je hodnota vstupní proměnné.

Mealy

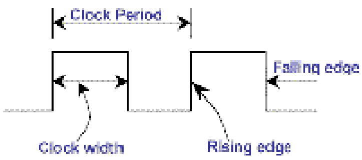
výstup závisí na vnitřním stavu i vstupech. Tabulka přechodu a výstupu obsahuje kromě přechodu pro jednotlivé vstupy X i výstup Y pro každý vstup X. Jeden vnitřní stav tedy má několik výstupních hodnot v závislosti na aktuálním vstupu. Když tento automat kreslíme, výstupní hodnota není v uzlu ale také na hraně.

Možno realizovat převody mezi automaty viz skripta LS strana 116 až 119. Převod Mealy na Moore: Nový automat bude mít více stavů. Převod Moore na Mealy: Změní se pouze tabulka výstupů (nyní bude závislá také na vstupech). Ta bude vypadat de facto stejně jako tabulka přechodů, pouze v ní budou výstupy odpovídající vnitřním stavům.

Synchronní a asynchronní sekvenční obvody a jejich návrh

Sekvenční obvody dělíme do dvou hlavních části: synchronní a asynchronní. Jejich klasifikace záleží na způsobu časování jejich signálů.

Synchronní sekvenční obvody mění svůj stav a výstupní hodnotu v diskrétních násobcích času, které jsou specifikovány nástupnou nebo sestupnou hranou jejich nezávisle běžícího hodinového signálu. Hodinový signál je generován z čtvercového signálu jako signál zobrazený na obrázku.



V asynchronních sekvenčních obvodech je změna stavu z jednoho na druhý iniciována změnou primárních vstupů, není zde žádná externí synchronizace. Paměť obvykle používána v asynchronních sekvenčních obvodech je časově zpožděná, obyčejně způsobena zpětnou vazbou mezi logickými hradly. Tudíž mohou být asynchronní sekvenční obvody považované za kombinační obvody se zpětnou vazbou. Kvůli zpětné vazbě mezi hradly, mohou být asynchronní sekvenční obvody občas nestabilní kvůli přechodným podmínkám. Tato nestabilita způsobuje návrháři mnohé problémy. Z toho důvodu se tolik nepoužívají jako synchronní obvody.

Synchronní obvod reaguje na hodiny (nástupná x sestupná hrana x obojí) v jinou dobu nereaguje na vstupy. Asynchronní obvody nejsou závislé na hodinových impulzech (asynchronní reset, atd.)

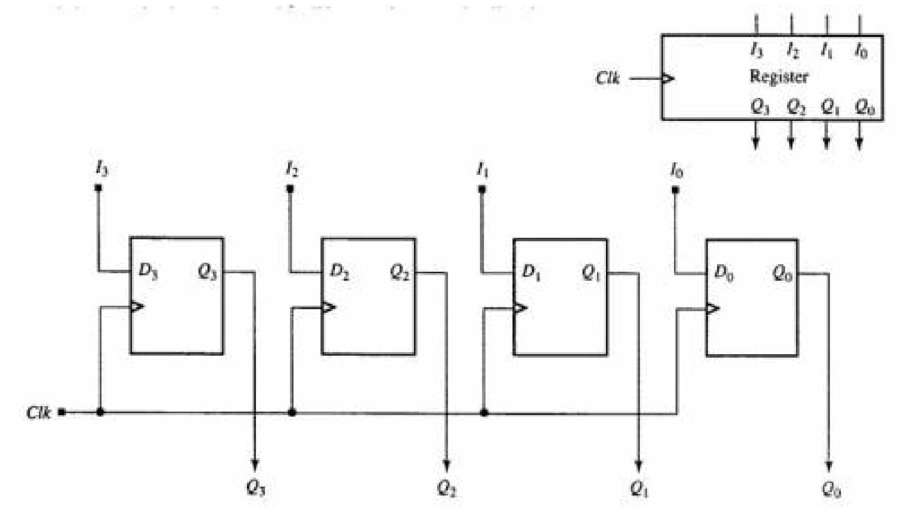
Čítače

Pro všechny elektronické čítače platí, že jsou schopny sečíst počet impulsů, které přicházejí na jejich vstupy, a tento počet uložit do paměti. Přitom je po vlastní čítání i pro jeho výsledek nepodstatné, jak je čístač technicky realizován, pokud platí, že vyhodnocované impulsy vyhovují požadavkům čítače nebo alespoň, že jsou těmto požadavkům vhodně přizpůsobeny. Na poměr mezi dobou impulsu a velikostí mezery mezi dvěma impulsy(střídu) nejsou kladeny žádné požadavky. Může jít tedy o periodické impulsy nebo o zcela nepravidelně časově rozložené impulsy. Moderní elektronické čítače pracují díky svým fyzikálním vlastnostem velice rychle, s frekvencí impulsů řádově do desítek MHz.

- *asynchronní* - u těchto čítačů je taktovací signál odvozen vždy z výstupu předchozího stupně (čistě asynchronní) nebo z některého z předchozích stupňů. Taktovací signál v sobě nese informaci o stavu předchozích stupňů a proto mají asynchronní čítače jednodušší zapojení ve srovnání se synchronními. Díky šíření signálu přes jednotlivé stupně zapojení vzikají v obvodu časová zpoždění, a tím i nežádoucí přechody stavů.
- *synchronní* - u těchto čítačů jsou všechny taktovací vstupy klopných obvodů připojeny na společný taktovací signál. Proto všechny klopné obvody reagují na stejnou hranu (náběžnou nebo sestupnou) taktovacího signálu. U těchto obvodů nevznikají na výstupech nežádoucí stavy. Co do zapojení jsou synchronní čítače složitější než obvody asynchronní, jsou však ve srovnání s asynchronními signály rychlejší.

Registry

n- klopných obvodů, řízených společným hodinovým signálem – zde příklad 4 bitového registru



Kódování vnitřních stavů v synchronních i v asynchronních systémech

Chceme-li automat realizovat, musíme jeho vnitřní stavy nějakým způsobem zakódovat tak, aby bylo možno tyto kódy uchovávat v klopných obvodech (vnitřní kód). V zásadě jde o to, jak vnitřní stavy zakódovat tak, aby výsledné kódování bylo realizovatelné co nejmenším počtem hradel, co nejjednodušeji.

Neexistují efektivní algoritmy, které by získaly optimální řešení, jsou však metody, které se tomuto řešení blíží.

Synchronní systémy

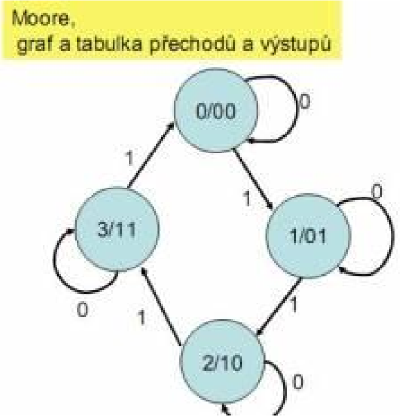
- synchronizační impulzy (náběžná/závěrná hrana hodinového impulsu)
- možná jakákoliv volba kódování, ale vedou na různě velké množství logiky - metoda Dolotta a McCluskey - vede na relativní optimum
- Metoda Dolotta and McCluskey - založena na kódotvorných sloupcích. Pro tyto sloupce jsou určeny podmínky –
- mají stejnou délku jako počet stavů, začínají nulou, obsahují stejné množství jedniček a nul (maximálně jedna jednička nebo nula navíc, když je počet stavů lichý). Vybíráme vhodné kódotvorné sloupce – bodují se atp.

Asynchronní systémy

- časové okamžiky se nazývají takty
- nejsou možná všechna zakódování - kódování podřízeno funkčnosti
- souběh - přechod mezi dvěma různymi vnitřními stavy, ve kterém dochází ke změne hodnot u

více než jedné vnitřní proměnné. Pokud pořadí těchto změn může ovlivnit provedení požadovaného přechodu, pak se jedná o kritický souběh, jinak je to nekritický souběh.

Návrh synchronních sekvenčních systémů



Q/E	0	1	Y
0	0	1	00
1	1	2	01
2	2	3	10
3	3	0	11

Použijeme příklad uvedený na slidech. Jedná se o návrh synchronního čítače M4 v binárním kódu (tzn., že výstupy budou 0,1,2,3,0,1….). Výstup tady není závislý na vstupech ale jen na vnitřním stavu (proto Moore).

Nejdříve si musíme sestavit graf a tabulku přechodů a výstupů. U grafu vidíme vždy stav – tj. 0/,1/… a výstup – tj. /00, /01… Jednička vždy způsobí přechod do dalšího stavu, nula nic. V tabulce vidíme co se stane v jednotlivých stavech (modrá barva) po příchodu 0 (nic) a 1 (přechod do dalšího stavu) a výstup (Y).

Teď je potřeba jednotlivé stavy zakódovat, tzn. přiřadit jim nějaký číslo (jelikož je to M4 – nejvyšší číslo je 11, vystačíme si s 2bitovým číslem.

q0 a q1 jsou jednotlivé složky stavu. E znamená to, co přijde. Všimněme si, že tabulka je stejná jako předchozí, jen místo 0,1,2,3 jsou tam binární ekvivalenty. Pro každou vstupní proměnou (q0,q1) je potřeba jedna výstupní funkce. My máme dvě, proto si musíme udělat dvě karnaughovy mapy. Jedna je pro Dq0 (hledám v červených číslíčkách pod velkými 0 a 1, kde nabývají hodnoty 1), to je třeba hned první řádek – q1 a q0 jsou tam nuly a E=1, takže hledám chlíveček (v karnaughově mapě), který není pod pruhem q0, q1 a je pod pruhem E – je to ten vlevo dole. A takhle úplně stejně se to udělá pro každou jedničku. Z karnaughovy mapy si zjednodušíme výstup a je téměř hotovo.

q_1q_0 \ E	0	1
00	00	01
01	01	10
10	10	11
11	11	00

0	1	1	0
1	0	0	1

q1, q0, E

$$D_{q_0} = q_0\overline{E} + \overline{q_0}E = q_0 \oplus E$$

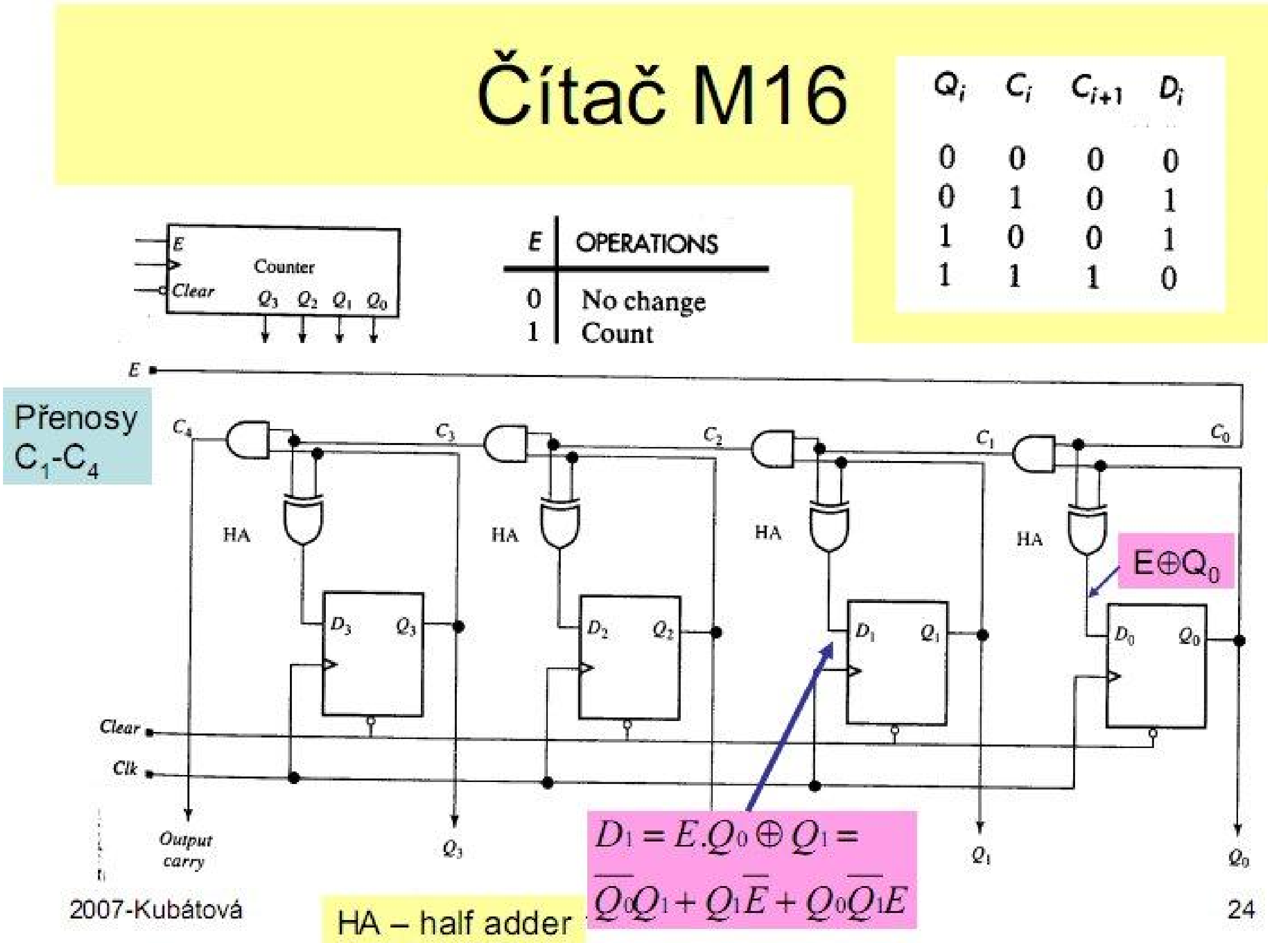
0	0	1	1
0	1	0	1

q1, q0, E

Výstupy: $Y_0 = q_0$, $Y_1 = q_1$

$$D_{q_1} = \overline{q_0}q_1 + q_1\overline{E} + q_0\overline{q_1}E$$

Na slidech je trochu nepochopitelně jako výsledek čítač m16, ale to v podstatě nevadí. Výsledkem by byla jen ta pravá polovina. Jelikož se jedná o sekvenční obvod, musí tu být ještě nějaká paměťová část, což jsou registry.



Zdroje

Číslicová technika, Marcela Antošová, Vratislav Davídek, KOPP 2003, ISBN 80-7232-206-0 Web předmětu Y36SAP (slidy, skripta) [http://service.felk.cvut.cz/courses/Y36SAP/]

CWUT - Logické funkce a formy jejich popisu [http://www.student.cvut.cz/cwut/index.php/Logick%C3%A9_funkce_a_formy_jejich_popisu]

Wikipedia:

Grayův kód (en) [http://en.wikipedia.org/wiki/Gray_code]

Karnaughova mapa (en) [http://en.wikipedia.org/wiki/Karnaugh_map]

Logický člen [http://cs.wikipedia.org/wiki/Logick%C3%BD_%C4%8Dlen]

Pravdivostní tabulka [http://en.wikipedia.org/wiki/Truth_table]

spolecne/spol10.txt · Poslední úprava: 2010/06/16 05:11 autor: Kucera5