

## 6.8 Čítače

Čítače tvoří samostatnou část sekvenčních logických obvodů a jak z názvu plyne, slouží k počítání. To co počítají jsou impulzy přiváděné na vstup a výsledek je pak na výstupu v příslušném kódu. Řešení čítačů je založeno na bistabilních klopných obvodech s případnou podporou kombinační logiky. Ta umožňuje vyřešit specifické požadavky na čítač. Základní dělení čítačů je na

- synchronní
- asynchronní

Další dělení podle směru čítání (vzestupně nebo sestupně) je na

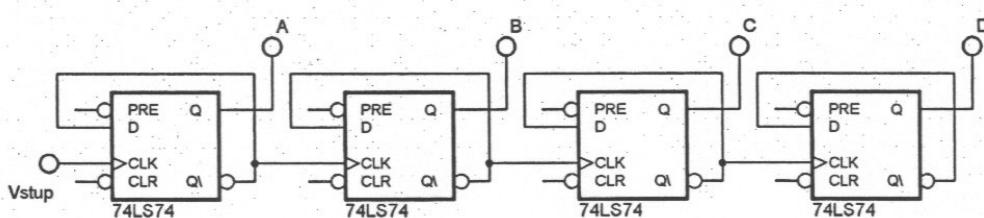
- vpřed
- vzad
- obousměrné

Důležité je dělení podle použitého kódu

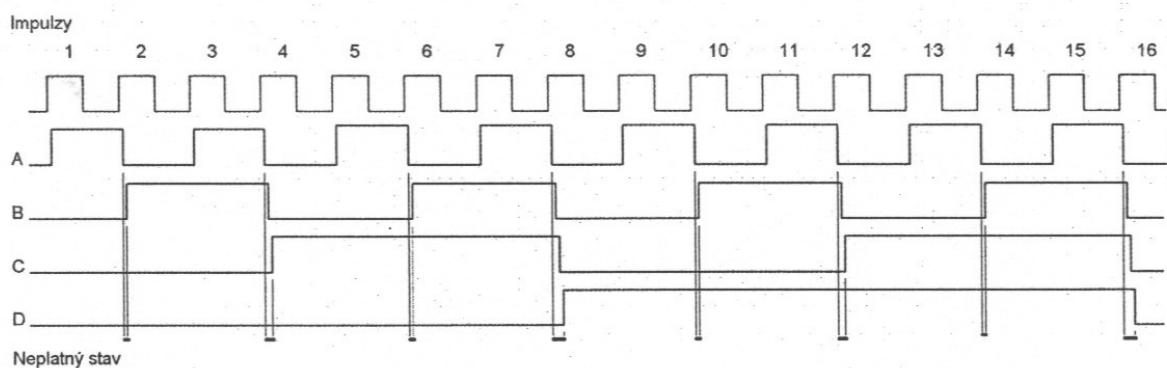
- binární
- dekadický
- Johnsonův
- a prakticky jakýkoliv jiný

Již dříve byly vysvětleny pojmy synchronní a asynchronní. V souvislosti s dělením čítačů na synchronní a asynchronní je však třeba upozornit, že kromě základního významu, je zde ještě jeden význam. Zda s příchodem počítaného impulzu dochází k překlopení všech klopných obvodů současně nebo postupně. U synchronních čítačů je to současně, tedy překlopení je synchronní se vstupním impulzem. U asynchronního čítače se s příchodem vstupního impulzu překlopí pouze klopný obvod nejnižšího řádu a postupně se přenáší do vyšších řádů. Z uvedeného popisu je zřejmá nevýhodnost asynchronních čítačů. Ta je dána skutečností, že údaj na výstupech čítače v době před ustálením je nesmyslný. Druhým nedostatkem je relativní pomalost oproti synchronním čítačům. Pokud vám není tato skutečnost zřejmá, zamyslete se nad následujícím. U synchronního čítače se s příchodem impulzu překlopí všechny klopné obvody současně. Příchod dalšího impulzu je tedy dán zpožděním jednoho klopného obvodu. Protože u asynchronního čítače je překlopení postupné, je doba do příchodu dalšího impulzu dána násobkem zpoždění jednoho klopného obvodu. Přičemž  $n$  je počet KO. Proto se asynchronní čítače prakticky nepoužívají. Jejich výhodou je totiž jen jednoduchost.

Takto je zapojen asynchronní binární čítač vpřed z klopných obvodů typu D



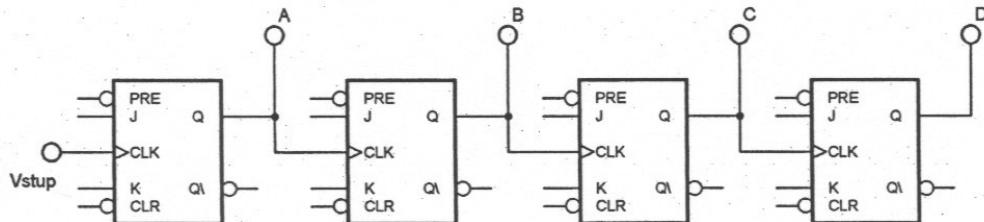
Jak je zřejmé, čítač je vlastně tvořen kaskádou jednobitových čítačů. Pro správné pochopení si uvědomte, že každý takový jednobitový čítač je vlastně dělička dvěma. Čítané impulzy přivádíme na první KO a na druhý KO pak s poloviční frekvencí atd. Na výstupech A, B, C a D pak obdržíme signály následujících průběhů



Jistě si uvědomujete, že taktovací impulzy jsou odebírány z invertovaného výstupu předchozího stupně. Proto k překlápení druhého a dalších stupňů dochází se závěrnou hranou impulzů na výstupech. Vlivem zpoždění, jak bylo dříve vysvětleno, jsou při každém druhém impulzu neplatné (nesmyslné) stavy na výstupu. Např. s příchodem čtvrtého impulzu dochází k následující posloupnosti přechodu stavů  $3 \rightarrow 2 \rightarrow 0 \rightarrow 4$ . Červenou barvou je opět vyznačen neplatný stav. Pravda je, že jsou některé aplikace, kde tato vlastnost nevadí, ale v naprosté většině případů je to nepřípustné. Pokud vás žádná aplikace, kde to nevadí nenapadá, zamyslete se nad čítačem, který bude počítat nahodilé impulzy a stav čítače budeme sledovat opticky. V tom případě dojde k přechodu přes neplatný stav tak rychle, že to vaše oko ani nepostřehne.

Je však třeba se ještě vrátit k zapojení tohoto čítače. Na schématu nejsou zapojeny nastavovací a nulovací vstupy. Ty je samozřejmě třeba ošetřit. Tyto vstupy musí být zapojeny na log 1, ale je možné i takové řešení, které umožní krátkodobé připojení aktivní úrovně. Tím by bylo možné vyřešit nulování nebo nastavení čítače, dokonce kombinací lze nastavit libovolnou hodnotu. Za zamyšlení jistě stojí taková změna zapojení čítače, kdy budou taktovací impulzy z přímého výstupu přechozího stupně. V této chvíli by již pro vás neměl být problém si odvodit průběhy impulzů na výstupech a zjistit, že se tak získá čítač vratný.

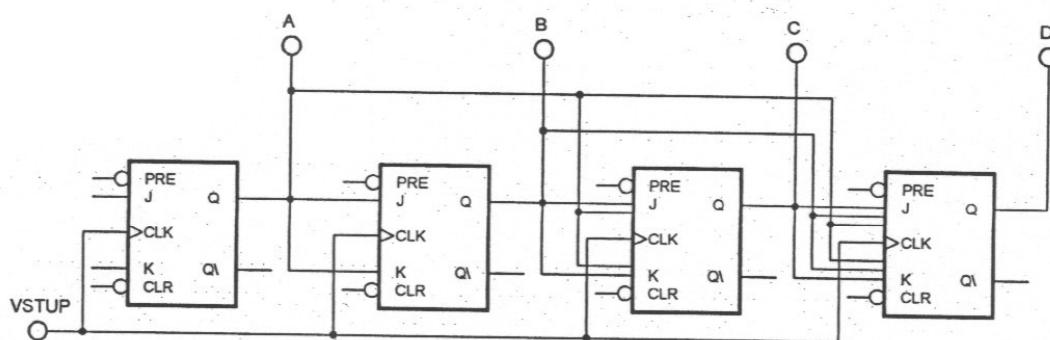
Druhou možností pro realizaci asynchronního binárního čítače vpřed, je použití klopných obvodů JK. Obrázek ukazuje zapojení.



Z pravdivostní tabulky obvodu JK plyne, že funguje jako dělička dvěma v případě, že na vstupech je log 1. Samozřejmě je třeba i zde ošetřit vstupy nastavení a nulování. Také změna na čítač vzad je analogická předchozím. Tedy taktovací vstupy budou zapojeny na invertované výstupy předchozích stupňů. Pokud se chcete přesvědčit, že jste všemu dobře porozuměli, nakreslete si graf průběhu signálů na výstupech podobně jak bylo uvedeno v předchozím. Nezapomeňte, že k přenosu signálu na výstup dochází se závěrnou hranou vstupního impulzu.

Asynchronní čítače mohou samozřejmě být i s jiným než binárním kódem. Prakticky však snad s výjimkou BCD dekadických čítačů nemají použití. Protože se v tomto případě jedná vlastně o zkrácený binární kód, spočívá řešení buď ve využití přídavné logiky nebo ve využití klopních obvodů JK, které mají oba vstupy rozšířeny jako AND hradla.

Zapojení synchronního binárního čítače vpřed je obvykle řešeno pomocí klopních obvodů typu JK. Jak je to zapojeno ukazuje následující obrázek.

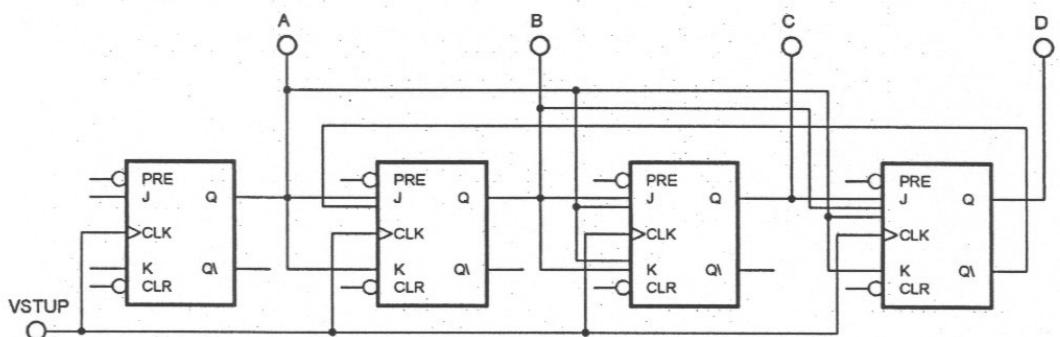


Jak vidíte, oproti předchozímu má tento čítač všechny hodinové vstupy navzájem propojeny a celkově je složitější. Je nutné, aby vstupy klopních obvodů byly řešeny jako hradla AND. Pravda je, že v době, kdy se čítače fyzicky realizovaly z jednotlivých klopních obvodů, byl vyráběn i vhodný klopní obvod s typovým označením 7472. Ten právě měl vstupy J a K řešeny jako třívstupová součinnová hradla. Dnes se již sice čítače realizují v integrované podobě (tedy celý čítač jako jeden integrovaný obvod), ale uvnitř najdete principálně stejně zapojení. Pokud se jedná o něco složitějšího je to proto, že obvod má zabudovány některé další funkce. Jistě si dovedete představit, že například nulování čítače je vyřešeno propojením všech vstupů CLR a vyvedením na jeden pin IO. Obdobně také funkce nastavení. To, že všechny vstupy musí být zapojeny na úroveň log 1, snad již není třeba ani připomínat.

Celou činnost tohoto čítače můžeme začít ve stavu, kdy výstupy budou na úrovni log 0. Jistě si uvědomujete, že A je LSB a D je MSB. Ze zapojení plyne, že s výjimkou prvního klopního obvodu, který má na vstupech log 1, mají všechny ostatní vstupy na log 0. Z pravdivostní tabulky klopního obvodu JK plyne, že s příchodem impulzu na vstup se překlopí pouze první obvod, tedy je započítán jeden impulz. Před příchodem druhého impulzu tedy mají na vstupy log 1 první dva klopné obvody. S příchodem impulzu se tedy právě tyto dva obvody překlopí. První na nulu, druhý na jedničku. Uvědomte si co to znamená. Čítač bude ve stavu ABCD = 0100, tedy dekadicky dva. Vstupy třetího obvodu

jsou logickým součinem výstupů prvních dvou, tedy log 0. Proto s příchodem dalšího impulzu překlopí pouze první klopný obvod a čítač je ve stavu dekadicky 3. Z uvedeného je zřejmé, že v tomto stavu s dalším impulzem překlopí první tři obvody. První dva překlápí do log 0 a třetí do log 1. Jistě si již dokážete odvodit všechny stavy, kterými čítač prochází při dalších impulzech na vstupu.

Často používaným čítačem je také dekadický čítač s kódem BCD. Jeho zapojení je na následujícím obrázku.



Při porovnání s binárním čítačem vidíte velkou podobnost. To je zcela pochopitelné, protože oba čítače až do stavu 1001 fungují stejně. U dekadického čítače však z tohoto stavu musí překlopit pouze první a poslední klopný obvod, u ostatních naopak musí být zajištěno, že nepřeklopí. K návrhu čítače se používají Karnaughovy mapy, ale je možné použít pouhou logickou úvahu.

U prvního a třetího je to jednoduché, protože v porovnání s binárním čítačem je situace zcela identická. Odpověď na to, jak zajistíme, aby druhý klopný obvod zůstal v log 0, najdeme v pravdivostní tabulce klopného obvodu JK. Z ní zjistíte, že na vstupech musíme zajistit  $J = 0$  a  $K = 1$ , ale pouze při zmíněném stavu 1001! Proto je na jeden vstup J druhého obvodu přiveden stav z invertovaného výstupu posledního klopného obvodu. Uvědomte si, že tímto propojením přivádíme úroveň log 1 teprve po započítání osmi impulzů, ale výstup prvního obvodu má úroveň log 0. Proto s příchodem devátého impulzu dojde k překlopení pouze prvního obvodu a stav čítače je tak skutečně devět. Rovněž pro zapojení vstupů posledního klopného obvodu využijeme pravdivostní tabulku, resp. vyjdeme z požadavku  $J = 0$  a  $K = 1$  při stavu devět. Toho je dosaženo tak, že zatímco výstup prvního hradla přivádíme na oba vstupy posledního obvodu, do logického součinu vstupu J přivádíme výstupy všech předchozích klopných obvodů. Tak je zajištěno, že při stavu 1110 je pro poslední obvod splněna podmínka k překlopení, tedy k přechodu čítače do stavu 0001. V této chvíli máme na obou vstupech posledního obvodu log 0 a s příchodem dalšího impulzu tedy může překlopit pouze první obvod, ale ostatní, včetně posledního, ne. Tím získáme stav 1001 tedy devět. Pokud se podíváte na zapojení vidíte, že podmínka pro nulování čítače ze stavu devět je splněna.

Možná vás v této chvíli napadlo řešení, že použijeme binární čítač s dekadérém stavu deset a výstup dekodéru přivedeme na nulovací vstup. A že by tento princip bylo možné použít pro čítače s různým rozsahem. Máte pravdu, fungovalo by to, ALE! Na výstupu by

se na krátkou dobu objevil stav deset (prostě o jedničku víc než maximální stav) a to by jistě byl hazardní stav. Navíc by se využitím asynchronního vstupu i vlastní čítač vlastně stal asynchronní.

Přesto však můžeme potřebovat čítač, který čítá do určité konkrétní hodnoty a pak se nuluje, aby se cyklus čítání opakoval. Takové čítače nazýváme modulo N. Za N pak doplňujeme konkrétní číslo. Často potřebným je čítač modulo 6. Jistě vás napadlo, že tento čítač budete potřebovat pro čítání desítek sekund a desítek minut v hodinách.

Pokud máme realizovat čítač modulo 6, ukážeme si postup návrhu pomocí map. Pro číslo šest potřebujeme tři bity a čítač bude nabývat stavů 000 až 011 (pozor na to co je LSB a co je MSB).

Při návrhu nejdříve vytvoříme tzv. logický plán. To znamená, že do jednotlivých polí mapy pro tři proměnné zapíšeme jejich dekadickou hodnotu. Čítač modulo šest nabývá hodnot 0 až 5, další hodnoty jsou nepřípustné. Logický plán pak je zapsán takto

		$B$		
		$A$		
C	0	1	3	2
	4	5	X	X

Pro oba vstupy všech tří klopných obvodů musíme vytvořit potřebné mapy, tedy šest map

$J_A$	$J_B$	$J_C$																								
$B$	$B$	$B$																								
$A$	$A$	$A$																								
<table border="1"> <tr> <td>1</td><td>X</td><td>X</td><td>1</td></tr> <tr> <td>C   1</td><td>X</td><td>X</td><td>X</td></tr> </table>	1	X	X	1	C   1	X	X	X	<table border="1"> <tr> <td>0</td><td>1</td><td>X</td><td>X</td></tr> <tr> <td>C   0</td><td>0</td><td>X</td><td>X</td></tr> </table>	0	1	X	X	C   0	0	X	X	<table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr> <td>C   0</td><td>X</td><td>X</td><td>X</td></tr> </table>	0	0	1	0	C   0	X	X	X
1	X	X	1																							
C   1	X	X	X																							
0	1	X	X																							
C   0	0	X	X																							
0	0	1	0																							
C   0	X	X	X																							

$K_A$	$K_B$	$K_C$																								
$B$	$B$	$B$																								
$A$	$A$	$A$																								
<table border="1"> <tr> <td>X</td><td>1</td><td>1</td><td>X</td></tr> <tr> <td>C   X</td><td>1</td><td>X</td><td>X</td></tr> </table>	X	1	1	X	C   X	1	X	X	<table border="1"> <tr> <td>X</td><td>X</td><td>1</td><td>0</td></tr> <tr> <td>C   X</td><td>X</td><td>X</td><td>X</td></tr> </table>	X	X	1	0	C   X	X	X	X	<table border="1"> <tr> <td>X</td><td>X</td><td>X</td><td>X</td></tr> <tr> <td>C   0</td><td>1</td><td>X</td><td>X</td></tr> </table>	X	X	X	X	C   0	1	X	X
X	1	1	X																							
C   X	1	X	X																							
X	X	1	0																							
C   X	X	X	X																							
X	X	X	X																							
C   0	1	X	X																							

Aby byl výsledek co nejjednodušší, je třeba v maximální míře využít neurčité stavy. Ty jsou samozřejmě u hodnot, které jsou nepřípustné. Pokud pochybujete uvědomte si, že

tyto stavy skutečně nemohou nastat a tudíž na stavu vstupů určitě nezáleží. Je však možné se zamyslet nad tím, jak je možné využít neurčitých stavů i na jiných místech tabulky. Odpověď opět spočívá v pravdivostní tabulce obvodu typu JK. V případě, že je výstup obvodu v log 0 je možné zajistit přechod do log 1 buď jako překlopení obvodu ( $J = 1, K = 1$ ), nebo jako nastavení obvodu ( $J = 1, K = 0$ ). Druhý uvažovaný případ vychází z nastaveného stavu výstupu. Přechod do nuly pak zajistíme překlopením ( $J = 1, K = 1$ ), nebo jako nulování obvodu ( $J = 0, K = 1$ ). V prvném tedy nezáleží na stavu K ( $K = X$ ), ve druhém případě na stavu J ( $J = X$ ). Uvedené bylo u map skutečně maximálně využito a výsledkem jsou následující vztahy

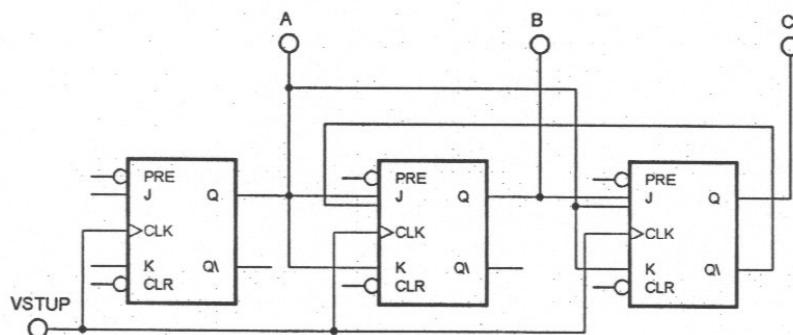
$$J_A = K_A = 1$$

$$J_B = A \cdot \bar{C}$$

$$K_B = K_C = A$$

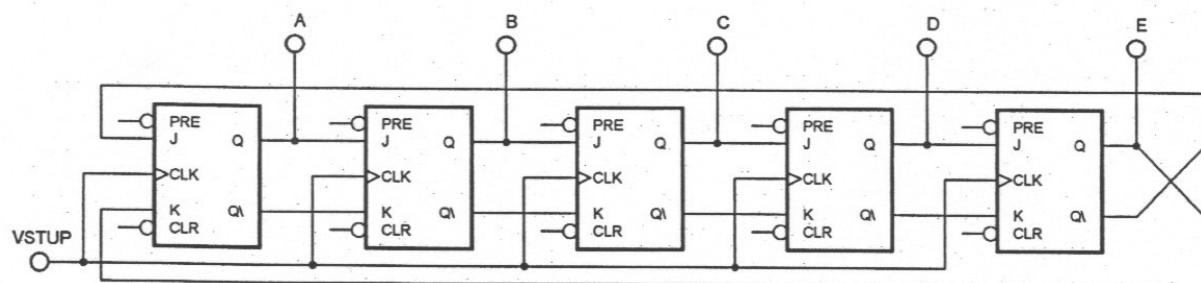
$$J_C = A \cdot B$$

Výsledné zapojení čítače modulo 6 pak ukazuje následující obrázek



Pokud budete někdy potřebovat čítač modulo 6, jistě použijete toto řešení, které ostatně najdete v literatuře běžně. Pokud budete potřebovat čítač modulo něco jiného, použijete tento naznačený postup. Stejně tak, pokud budete navrhovat čítač s jiným kódem.

Zajímavým čítačem je ten, který pracuje s Johnsonovým kódem. Tento kód se vyznačuje tím, při změně stavu o jedničku se mění úroveň pouze jednoho bitu. Do značné míry připomíná posuvný registr. Prakticky to znamená, že desítkový čítač potřebuje k zobrazení pět bitů. Čítač je zapojen následujícím způsobem.



Vlastní kód pak má následující podobu

<i>dek.</i>	<i>EDCBA</i>
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

Z tabulky je zřejmé, proč tento čítač připomíná posuvný registr.

Již dříve bylo uvedeno, že čítače existují i v integrované podobě. Důvod je zřejmý. Zapojit čítač z klopňých obvodů je sice relativně jednoduché, ale neekonomické. A když už technologie nabízí tuto možnost tak proč ji nevyužít. Samozřejmě není možné vyrábět celý myslitelný sortiment, ale přesto ty nejobvyklejší jsou k dispozici, případně jsou udělány tak, aby poskytovaly více možností.

První typy čítačů, které dosáhly masivního rozšíření jsou asynchronní čítače typu 7490 a 7493, později samozřejmě obsažené i v dalších řadách.

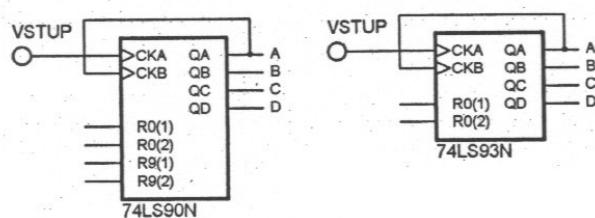
Čítač 7490 lze charakterizovat jako

- asynchronní BCD desítkový čítač
- s asynchronními nulovacími vstupy R0 a nastavovacími vstupy R9
- dělený na dvě části – dělička dvěma a dělička pěti

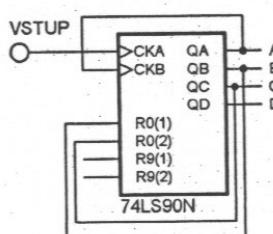
Čítač 7493 lze charakterizovat jako

- asynchronní binární čítač
- s asynchronními nulovacími vstupy R0
- dělený na dvě části – dělička dvěma a dělička osmi

Povšiměte si, že oba typy čítačů se skládají ze dvou nezávislých částí. Nejdříve je to dělička dvěma se vstupem CKA a výstupem A. Pak následuje dělička pěti, resp. osmi se vstupem CKB a výstupy B, C, D. Pokud je tedy chceme použít v základním zapojení, musíme propojit výstup A na vstup druhé části tedy vstup CKB. To ukazuje i následující obrázek.



V případě, že potřebujeme zkrátit cyklus čítače, lze použít asynchronní nulovací vstupy. Např. pro čítač modulo 6 lze odvodit nulovací signál od dvou jedniček v binárním vyjádření čísla 6.



Samozřejmě na konci čítací sekvence nastane na krátký okamžik stav 6, tedy hazardní stav, který vyvolá vynulování čítače. Jenže uvědomte si, že se stejně jedná o asynchronní čítač, který se vyznačuje hazardními stavami a tak vlastně nic nepokazíme.

Pokud naopak potřebujeme děličku s větším dělícím poměrem, resp. čítač s konečnou hodnotou vyšší než 15, můžeme tyto čítače řadit do kaskády. To v podstatě znamená, že výstup D propojíme se vstupem CKA (CKB) následujícího obvodu a tímto jednoduchým propojením jsme získali osmi (sedmi) bitový asynchronní čítač vpřed. I pro tuto kaskádu pak můžeme zapojit zkrácení cyklu podle výše uvedeného principu.

Další čítače, které dosáhly velké obliby jsou typy 74192 a 74193.

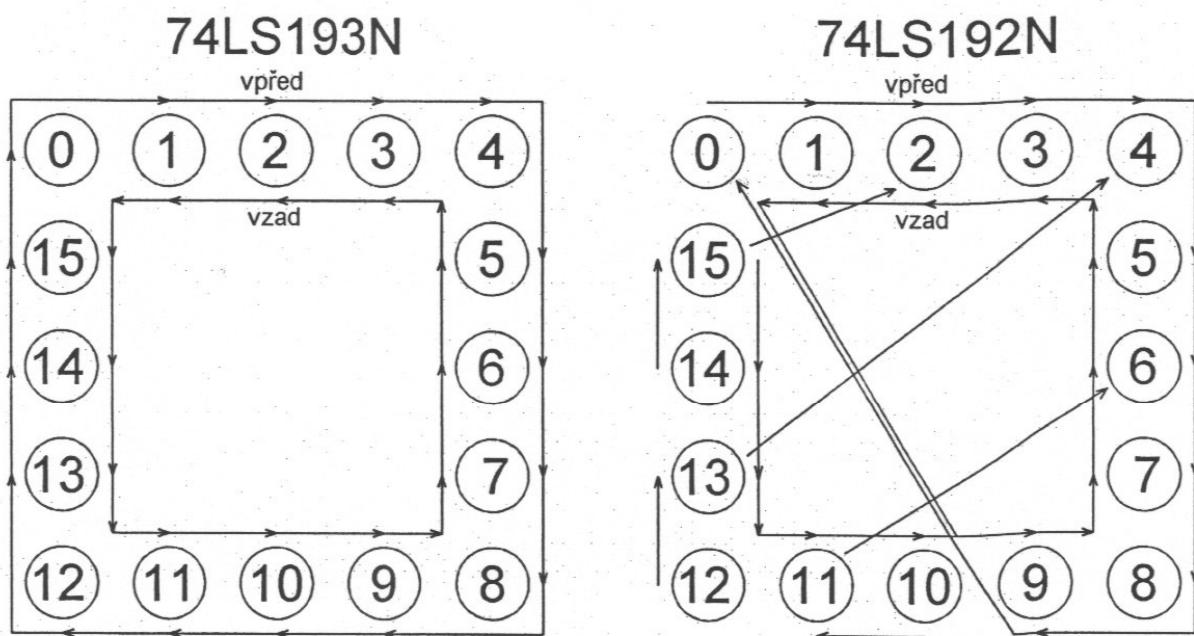
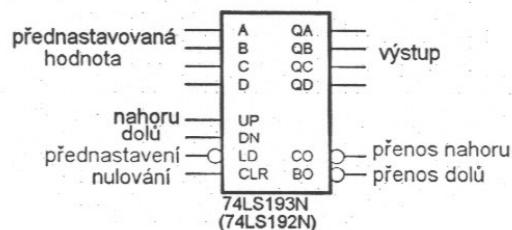
Čítač 74192 lze charakterizovat jako

- obousměrný, synchronní, BCD dekadický čítač
- s asynchronním nulováním
- s asynchronním přednastavením libovolného stavu
- s výstupy pro řazení čítačů do kaskády

Čítač 74193 lze charakterizovat jako

- obousměrný, synchronní, binární čítač
- s asynchronním nulováním
- s asynchronním přednastavením libovolného stavu
- s výstupy pro řazení čítačů do kaskády

Následující obrázek ukazuje schematickou značku a průběhy jednotlivých cyklů při čítání nahoru a dolů.



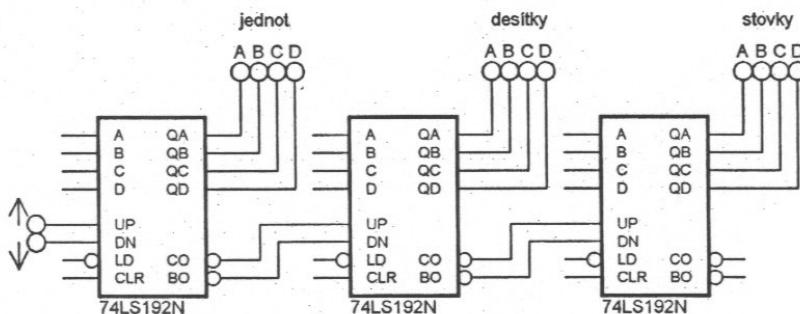
Oba typy čítačů mají dva vstupy, pro čítání nahoru a pro čítání dolů. Oba tyto vstupy reagují na náběžnou hranu impulzů. Výstupy přenosů umožňují řazení do kaskády a jsou aktivní v nule. To znamená, že v klidovém stavu je na těchto výstupech úroveň log 1. Přechodem do log 0 a zpět je tak vytvořen impulz pro následující stupeň.

Pro vstup nastavení platí, že je aktivní v log1 a v klidu je tedy třeba zabezpečit stav log 0.

Režim nastavení je u těchto čítačů velmi zajímavý, protože umožňuje nastavení libovolné hodnoty. Prakticky to vše probíhá tak, že na vstupy přednastavení přivedeme požadovanou hodnotu a pak pomocí impulzu s úrovní log 0 na vstupu přednastavení proběhne vlastní zápis. U obvodu 74LS193N není s přednastavením žádný problém. Prostě se čítá od přednastavené hodnoty. U obvodu 74LS192N je však situace složitější. Pokud je přednastavená hodnota v rozsahu 0 až 9, žádná komplikace nenastane. Pokud je však přednastavena hodnota v rozsahu 10 až 15, bude se čítač při čítání vpřed chovat podle obrázku. Pravda je, že čítač je dekadický a zmiňovaný problém by tedy z principu měl být vyloučen, ale přesto mějte uvedenou skutečnost na paměti.

Pozor i na nevyužité vstupy, resp. jejich ošetření.

Následující obrázek ukazuje, jak jednoduše lze čítače této řady použít.



Jedná se o obousměrný čítač v rozsahu 0 až 999, tedy tří dekád. Pokud by vám to bylo málo, lze jednoduše přidat další obvody. Snad není třeba připomínat, že je třeba ošetřit zde nevyužité vstupy nulování a předenastavení.

## 6.9 Hradlová pole druhé a třetí generace

Hradlová pole, tak jak byla představena ve své první generaci, se samozřejmě stala základem pro vývoj druhé generace. Pro ni je typické použití tzv. makrobuněk. Do nich bylo převzato všechno co nabízela výchozí základna, dokonce zde byla snaha o určitou kompatibilitu. Jednalo se o obvody typu GAL 16V8 a GAL 20V8, které ve své době nahradily všechny obvody PAL z první generace. Protože však právě přílišná vazba na první generaci výrazně omezovala aplikovatelnost těchto obvodů, nutně musel následovat vývojový skok. Proto přišly obvody GAL 22V10, které přinesly architekturu pro univerzální použití. Obvod má 22 vstupů AND, 10 výstupů a jeho typickým prvkem je tzv. makrobuňka (OLMC - Output Logic Macro Cell). Makrobuňka umožňuje vytvořit kombinační i registrové funkce. V každé makrobuňce může být vytvořena logická funkce s až 16 mintermy. Uvědomte si, že každému výstupu odpovídá jedna makrobuňka. Makrobuňky však lze slučovat, takže pokud nevadí omezení počtu funkcí, může mít jedna funkce i více mintermů. Registrové funkce jsou zapojeny až za kombinační částí a umožňují vytvářet posuvné registry, čítače atd. Přesto, že jsou tyto obvody dnes velmi používány, budoucnost patří třetí generaci hradlových polí.

Ve třetí generaci jsou to obvody CPLD a FPGA.

Obvody CPLD - Complex Programmable Logic Devices jsou přímým pokračovatelem vývoje hradlových polí druhé generace. I zde je hlavním stavebním kamenem makrobuňka typu PAL nebo PLA. Tyto makrobuňky nejsou zcela nezávislé, ale sdružují se do bloků. Těch bývá obvykle dva až šestnáct. Tyto bloky jsou relativně nezávislé a jsou propojeny programovatelnou strukturou. Makrobuňky mohou fungovat jako skryté, tedy jejich výstupy nejsou vyvedeny na výstupy, ale tvoří vnitřní signál. Kromě toho tyto obvody obsahují další prvky rozšiřující možnosti použití. Jsou to obvody typu D, T, EX-OR, klopné obvody typu latch, případně obvody pro aritmetické operace. Jednotlivé