

1 Monolitické počítače

Jak ze základní architektury počítače víme, skládá se počítač ze tří částí: procesor (CPU), paměť a periférie. Pokud některá z těchto tří částí chybí, nejde o počítač.

Pro řadu nenáročných aplikací se již skoro 30 let vyrábějí malé počítače integrované v jediném pouzdře. Tvoří tedy jeden celek - monolit. Odtud tedy obecně používaný a zažitý název - **monolitické počítače**.

V praxi se ale často setkáváme s řadou dalších názvů, jako: jednočip, mikročip, mikrokontrolér a mikropočítač. Jde o názvosloví odvozené od zvyklostí různých výrobců a zde budeme používat i poslední uvedené.

1.1 Architektura monolitických počítačů

Známe dvě základní koncepce počítače: von Neumannovu a harvardskou.

Pro monolitické počítače se převážně využívá harvardská architektura, protože přináší důležitou vlastnost: oddělení paměti pro data od paměti programu. To umožňuje snadno konstruovat počítač, který má paměť dat i programu vyrobenou jinou technologií a navíc dovoluje mít odlišnou velikost nejmenší adresovací jednotky. Např. data jsou obvykle uložena po bytech, ale v případě signálových procesorů mohou být po slovech velikosti 16 bitů. A strojové instrukce procesoru mohou mít u každého procesoru jinou velikost, např. 12, 14, 16, 24 nebo 32 bitů a nejmenší adresní jednotka paměti programu se této velikosti přizpůsobuje.

Další dělení počítačů je obvykle podle použitého procesoru, tedy CISC a RISC. V dnešní době kvůli jednoduchosti nesou monolitické počítače převážně rysy architektury RISC, i když často ve velmi zjednodušené formě.

I přes určité společné rysy monolitických počítačů, nemůžeme jejich architekturu zobecnit a tyto rysy generalizovat. Každý výrobce nabízí počítače vhodné pro jiné aplikace a proto je třeba očekávat poměrně velké rozdíly mezi produkty jednotlivých výrobců.

1.2 Paměti monolitických počítačů

V monolitických počítačích můžeme najít dva základní typy paměti: pro data a pro program.

Pro data používáme většinou paměti energeticky závislé typu RWM-RAM (Read-Write Memory - Random Access Memory), tedy paměť s náhodným přístupem určenou pro čtení i zápis. Tyto paměti jsou vyráběny jako statické, jejich paměťové buňky jsou realizovány jako klopné obvody.

Pro program se používají paměti, které si svůj obsah zachovávají i po odpojení napájení, tedy jde o paměti typu ROM. Nejčastěji se dnes používají

paměti EPROM a Flash EEPROM. Nesmíme také zapomenout na výrobky s pamětí PROM, dnes označovanou spíše jako OTP (One Time Programmable).

Některé počítače mají označení „ROM-Less“ a nejsou osazeny vnitřní pamětí ROM. Tu je třeba s potřebným programem k počítači připojit jako externí.

1.3 Organizace paměti dat

Podle účelu, k jakému jednotlivé paměťové buňky slouží, můžeme v monolitických počítačích nalézt 3 základní typy pamětí:

střadačové (pracovní) registry - ve struktuře procesoru jsou obvykle jen jeden nebo dva. Ukládají se do nich aktuálně zpracovávaná data a jsou nejčastějším operandem strojových instrukcí. A také se do nich nejčastěji ukládají výsledky operací. Nejsou určeny pro dlouhodobé ukládání dat.

univerzální zápisníkové registry - slouží pro ukládání nejčastěji používaných dat. Instrukční soubor obvykle dovoluje, aby se část strojových instrukcí prováděla přímo s těmito registry. Formát strojových instrukcí ovšem obvykle nedovoluje adresovat velký rozsah registrů. Metodou, jak se omezení počtu zápisníkových registrů vyhnout, je implementace několika stejných skupin registrů vedle sebe, s možností mezi skupinami přepínat. Těmito skupinám říkáme **registrové banky**.

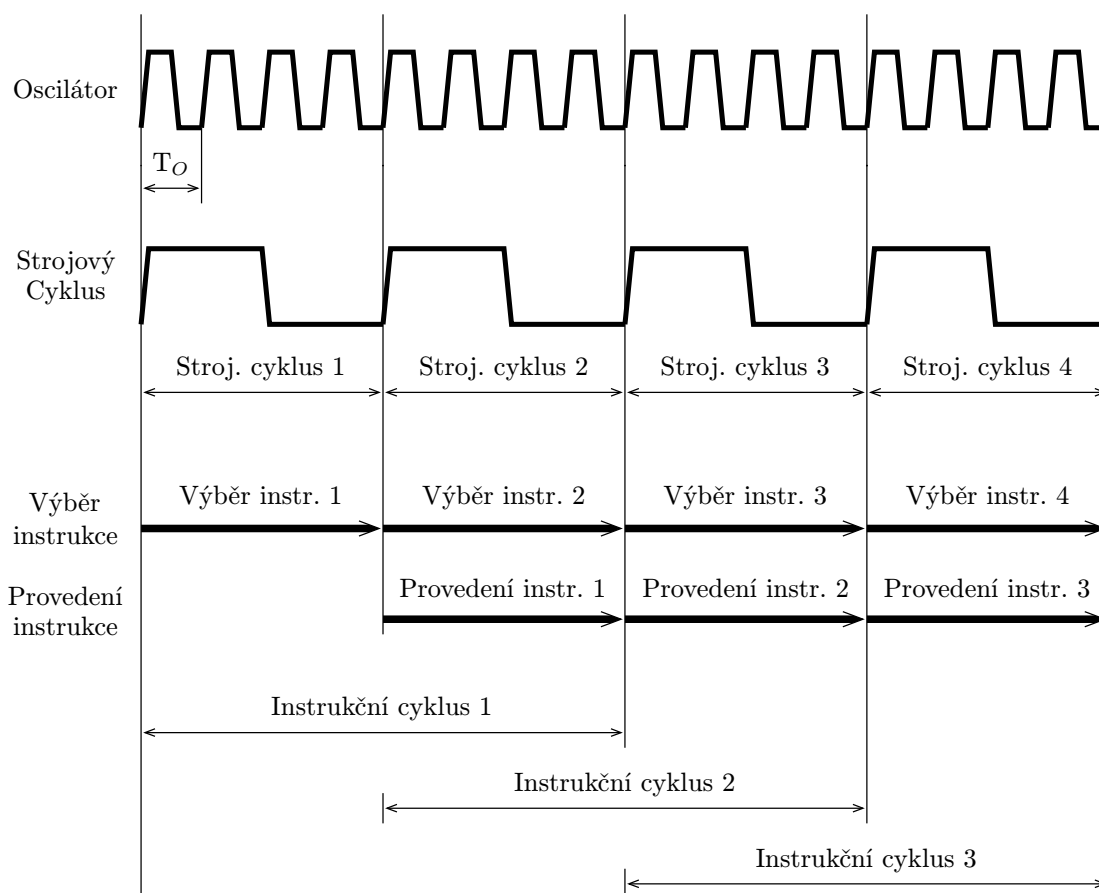
paměť dat RWM - slouží pro ukládání rozsáhlejších nebo méně používaných dat. Instrukční soubor obvykle nedovoluje s obsahem této paměti přímo manipulovat, kromě instrukcí přesunových. Těmi se data přesunou např. do pracovního registru. Některé procesory dovolují, aby data z této paměti byla použita jako druhý operand strojové instrukce, výsledek ale nelze zpět do této paměti uložit přímo.

Dále jsou v počítači i **speciální registry**. V této skupině určitě najdeme **čítač instrukcí**, ukazující na právě vykonávanou instrukci, která je v **instrukčním registru**.

Pro ukládání návratových adres slouží **zásobník**. Ten může být realizován přímo v datové paměti na přesně vyhrazeném místě, nebo jako speciální paměť LIFO, jejíž obsah nelze programově číst. V každém případě k zásobníku musí být ještě jednoúčelový registr - **ukazatel vrcholu zásobníku**. Hloubka zásobníku bývá často omezena u mikropočítačů na 2 až 8 úrovní.

1.4 Zdroj synchronizace

Počítače, stejně jako všechny sekvenční obvody, vyžadují pro svou činnost zdroj synchronizace. Ten vytváří časovou základnu pro provádění všech ope-

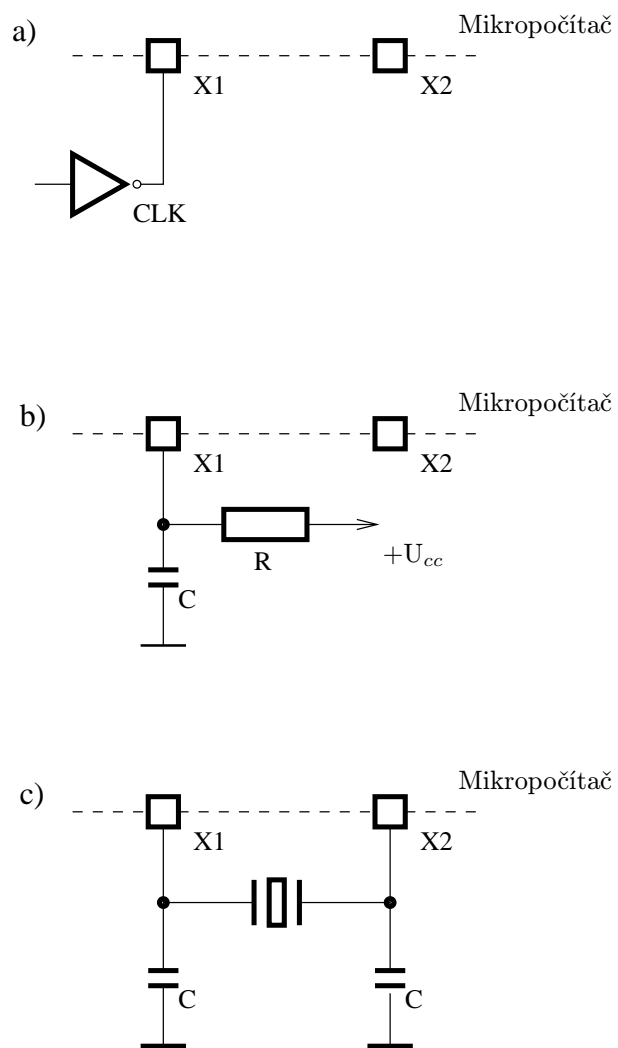


Obrázek 1: Příklad časových závislostí v počítači

rací v počítači. Příklad přepočtu hodinového cyklu na cyklus strojový a instrukční, je na obrázku 1.

Jde o příklad procesoru, kde se strojový cyklus vypočítá z hodinového dělením čtyřmi. Pro výběr strojové instrukce je třeba právě jeden strojový cyklus, stejně jako pro její vykonání. Instrukční cyklus tedy trvá 2 strojové cykly. Díky principu zřetězeného zpracování, kdy výběr následující instrukce probíhá současně s prováděním aktuální instrukce, je v každém strojovém cyklu dokončeno vykonávání jedné instrukce. Ovšem pokud nedojde k výpadku výběru instrukce, např. při skoku.

Často je tento zdroj integrován přímo v počítači, aby byl zajištěn co nejmenší počet externích součástek. Pokud je ale zdroj synchronizačního signálu integrován přímo na čipu, není možné tímto miniaturním provedením zajistit jeho dobrou stabilitu. Vlivem teploty mohou být odchylky kmitočtu i desítky procent. Toto provedení je tedy vhodné pouze tam, kde není nutná přesná doba vykonávání instrukce a není potřebná vazba na reálný čas.



Obrázek 2: Zdroje synchronizace a) externí zdroj b) RC c) s krystalem

Častěji se ale používají generátory, umožňující uživateli zvolit kmitočet pro řízení mikropočítače. Pro připojení časovacích prvků k počítači, jsou na pouzdře určeny jeden nebo dva vývody. Jako externí časovací prvek generátoru hodinového signálu se obvykle používá:

- krystal (křemenný výbrus),
- keramický rezonátor,
- obvod LC,
- obvod RC.

Pro dobrou stabilitu je vhodný krystal nebo keramický rezonátor - obr. 2b. Krystaly se vyrábějí v široké řadě hodnot a provedení. Uživatel jen musí vybrat správnou hodnotu z rozsahu hodnot doporučených výrobcem počítače. Pro běžné aplikace se např. volí kmitočty v rozsahu 1 až 20 MHz. Pro aplikace zálohované z bateriových zdrojů např. hodnoty v rozsahu 32 až 100 kHz.

U zařízení, kde je požadavek na minimalizaci ceny, jsou výhodné RC oscilátory - obr. 2c. Kmitočet se ovšem mění nejen s teplotou, ale i s velikostí napájecího napětí.

Při volbě řídicí frekvence je třeba mít ale vždy na paměti, že u obvodů vyrobených technologií CMOS, a těch je dnes většina, je celkový proudový odběr závislý na řídicí frekvenci velmi významným způsobem. Proto je třeba vždy zvážit, jaké jsou skutečné nároky na výpočetní výkon počítače nasazeného v konkrétní aplikaci.

Řada mikropočítačů je proto vybavena speciálními režimy, označovanými obvykle jako IDLE, SLEEP a HALT, kdy se mikropočítače přepínají do úsporného režimu instrukcí se shodným názvem jako uvedené režimy. V těchto režimech se snižuje hodinový kmitočet, nebo se hodiny úplně zastaví.

1.5 RESET monolitického počítače

Mikropočítač je sekvenční obvod. Jeho chování je proto závislé nejen na strojových instrukcích a jejich argumentech, ale i na stavech, které těmto instrukcím předcházejí a na signálech, které do tohoto obvodu vstupují.

Proto, aby bylo možno spolehlivě spustit běh programu, musí být přesně definován počáteční stav počítače. Ten je v technické dokumentaci označován jako tzv. „RESET“. K tomuto účelu jsou v mikropočítači implementovány inicializační obvody, které tento stav zaručí.

Po provedení RESETu je u všech počítačů nastaven počáteční hodnota čítače instrukcí. Obvykle je to hodnota 0, nebo samé jedničky.

Ve srovnání s klasickým sekvenčním obvodem je ale mikropočítač tvořen mnoha navazujícími obvody. Proto je RESET tvořen sérií navazujících činností, které musí inicializovat nejen registry a procesor, ale i periferní

obvody. Proto výrobce definuje, jakou dobu trvání musí mít signál RESET, aby byla inicializace provedena správně.

Zdroj signálu RESET může být vnější, nebo vnitřní. Vnější je obvykle realizován tlačítkem s RC obvodem, aby bylo zajištěno správné časování. I když je toto zapojení velmi jednoduché, je nutno mu věnovat patřičnou pozornost v souladu s dokumentací výrobce. Pro potřebu korektní inicializace při připojení napájení, je mikropočítač vybaven vnitřním zdrojem RESETu, kdy se např. Schmittovým klopným obvodem detekuje korektní úroveň napájecího napětí a zabezpečí se tak správná inicializace i v případě pomalého náběhu napájecího napětí.

1.6 Ochrana proti rušení

Tato problematika je zejména u mikropočítačů velmi důležitá. Mikropočítače jsou aplikovány v různých prostředích a ne vždy je okolí pro mikropočítač natolik přívětivé, abychom ho nemuseli chránit.

Na první místě většinou jde o ochranu mechanickou. Mikropočítač musí být vyroben tak, aby jej bylo možno za vývody bezpečně připojit na desku, bez nutnosti dalšího mechanického zabezpečení. V mnoha případech musí mikropočítač odolávat náhodným rázům, nebo i trvalým vibracím.

V mnoha případech je mikropočítač vystaven i elektromagnetickým vlivům z okolí. Pronikání těchto vlivů do mikropočítače může mít za následek nejen nestabilitu, ale i zničení obvodu. Proto se obvody musí v aplikaci odstínit a jednotlivé vývody i případně galvanicky oddělit od okolí.

A v neposlední řadě může jít o chyby samotného programátora, nebo nahodilé rušivé vlivy z okolí. Samozřejmě, čím déle počítač poběží, tím je větší pravděpodobnost, že se vliv okolí na mikropočítači projeví.

Vykonávaný program je určitá sekvence instrukcí, která se na základě stavu mikropočítače okolí opakuje. Vlivem okolí se ale může stát, že program občas „zabloudí“. Pro odstranění podobných situací je v mikropočítačích implementován speciální obvod nazývaný **WATCHDOG**.

Jde o samostatně běžící časovač, který svým přetečením, či podtečením, provede pomocí vnitřního RESETu reinicializaci mikropočítače. Pro vynulování čítače se používá obvykle speciální instrukce mikropočítače. Pokud je tedy tato instrukce pravidelně prováděna řídicím programem, k reinicializaci mikropočítače nedojde.

Dalším účinným způsobem ochrany před rušivými vlivy, je větší „**rozsah pracovního napětí**“. Pokud je rozsah např. 3 až 6 V a standardní napájecí napětí 5 V klesne i pod 4 V, nedojde v mikropočítači k žádným nežádoucím změnám hodnot v registrech a nedojde ke kolizi programu.

Často se u monolitických počítačů také setkáme se speciálním obvodem, který hlídá pokles napájecího napětí pod dovolenou úroveň. Pokud k takovému poklesu dojde, automaticky se provádí RESET. Tento obvod se často řeší jako pomocný obvod připojený na externí RESET signál mikropočítače.

1.7 Přerušovací podsystem

V dnešní době jsou procesory vybaveny přerušovacím podsystemem, který dovoluje výrazně efektivnější programové odezvy na změny stavu periférií a chování okolí.

Je třeba si uvědomit, že požadavky na přerušování přicházejí zcela asynchronně s činností hlavního programu. Proto může dojít k přerušování programu kdykoliv. Řízení procesoru se přeneslo definovaným způsobem na obsluhu přerušování, kde se musí v co nejkratším čase zajistit reakce na požadovaný podnět a vrátit řízení zpět do hlavního programu. Přitom obsluha přerušování musí uložit všechny systémové registry a na konci je vrátit do původního stavu, aby neměla žádný nežádoucí vliv na chod hlavního programu.

Pro programátora je důležité zajímat se, jak **povolit a zakázat přerušování**, jak detekovat, co je **zdroj požadavku na přerušování** a co musí povinně provést při obsluze přerušování (je to dáno výrobcem). Na některých mikropočítačích si také může určovat **prioritu** jednotlivých zdrojů přerušování a také je možné **vnoření** obsluhy více přerušování současně. Možnost vnoření bývá někdy kritická, protože může vést k rekurzivnímu zpracování téhož přerušování a při velmi omezené velikosti zásobníku pro návratové adresy je tento mechanismus nežádoucí.

1.8 Periférie - obvody vstupu a výstupu

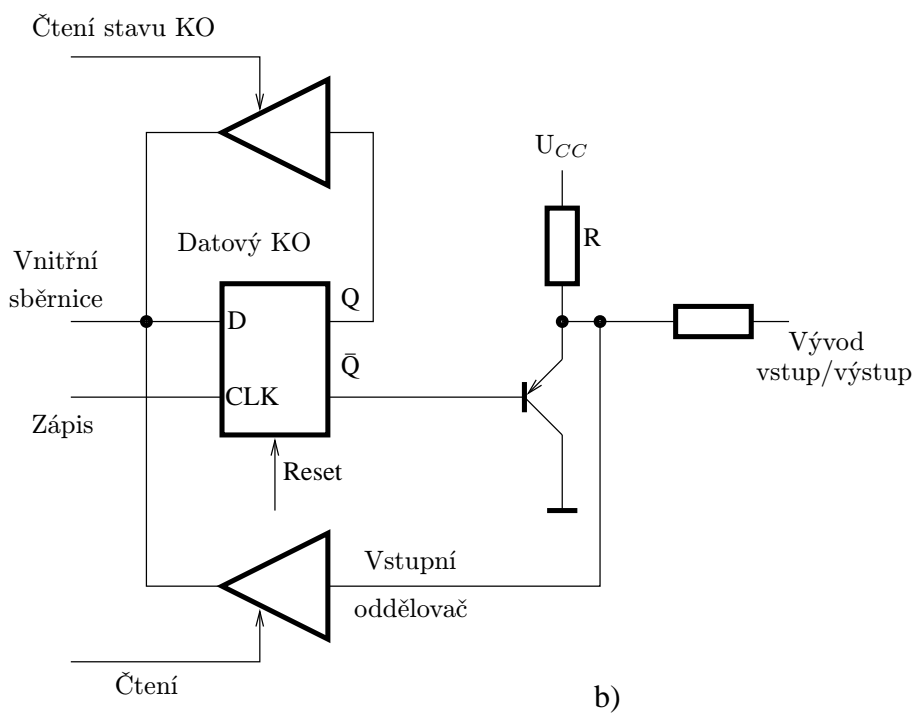
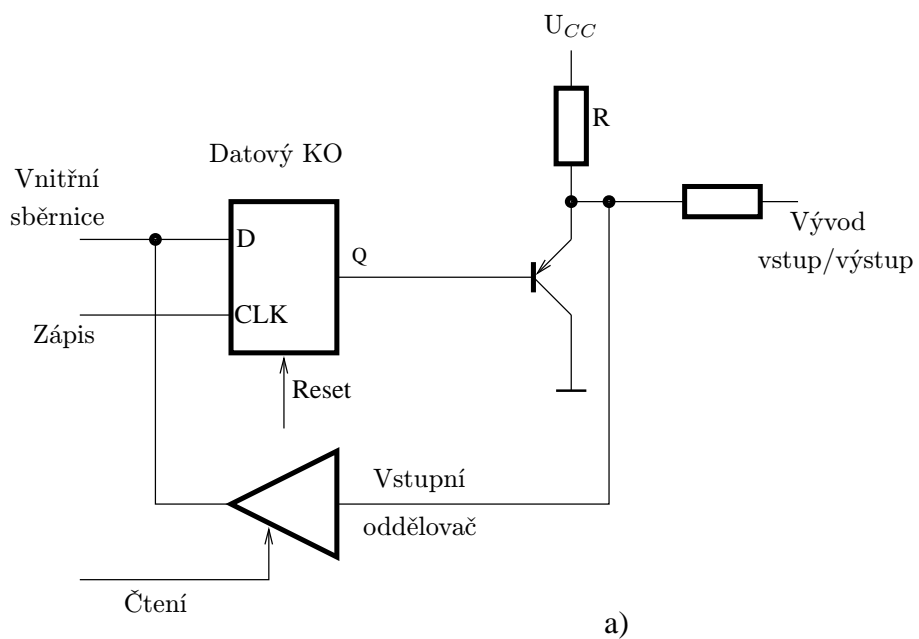
Obvody, které zajišťují komunikaci mikropočítače s okolím, nazýváme perifériemi. Jejich kvalita, množství a možnost snadného ovládnutí, určují celkovou výkonnost systému. Pokud obvod například potřebuje komunikovat po sériové lince a mikropočítač není tímto obvodem vybaven, musíme komunikaci emulovat.

1.8.1 Vstupní a výstupní brány

Nejjednodušší a nejčastěji používané rozhraní pro vstup a výstup informací je u mikropočítačů **paralelní brána - port**. Bývá obvykle organizována jako 4 nebo 8 jednobitové vývody, kde lze současně zapisovat i číst logické informace 0 a 1. U většiny bran lze jednotlivě **nastavit**, které bitové vývody budou sloužit jako **vstupní** a které jako **výstupní**.

U mnoha mikropočítačů jsou brány implementovány tak, že s nimi instrukční soubor může pracovat jako s množinou vývodů, nebo jako s jednotlivými bity.

Rozsah použití těchto univerzálních vstupů a výstupů je velmi široký. Slouží nejen pro čtení hodnot dvoustavových snímačů a ovládnutí výkonných přepínačů, ale lze jimi vytvořit i složitější rozhraní. Například komunikovat po sériové synchronní, nebo asynchronní lince běžně používanými rychlostmi. Podobně lze dynamicky ovládat LCD nebo LED displeje.



Obrázek 3: Struktura vstupně-výstupních bran - a) otevřený kolektor
b) s odděleným KO

Obecně platí, že vhodné programové vybavení může nahradit část speciálních obvodů. Musíme ale zvážit, zda procesorová část disponuje dostatečným výpočetním výkonem, aby zvládla programovou náhradu příslušné periférie.

Vnitřní strukturu obousměrných vstupně-výstupních vývodů ukazuje obrázek 3. K řízení směru přenosu se používá způsob známý ze sběrníkových systémů číslicových počítačů.

První způsob je analogické spojování logických obvodů s otevřeným kolektorem - obr. 3a. Na řízení směru komunikace není nutný samostatný registr. Zapsání logické „1“ do výstupního datového klopného obvodu nesepe tranzistor, a vývod je možno použít jako vstupní. Obvod na obr. 3b je podobný. Liší se jen tím, že data zpětně čtená po zapsání do záchytného registru nemusí být stejná, jako data čtená ze vstupu.

Další příklad na obr. 3c ukazuje zapojení s tranzistory N a P. Přepínání směru se řeší speciálním klopným obvodem, který povoluje sepnutí výstupních tranzistorů. U obousměrných bran musí vývojáři mikropočítače zajistit, aby se proti sobě neobjevily dva výstupy s opačnou logickou úrovní. Proto jsou obvykle obousměrné brány během RESETu nastaveny do vstupního režimu.

Kromě uvedených příkladů zapojení, mohou být v případě na obr. 3c zapojeny tzv. „pull-up“ nebo „pull-down“ odpory, které zajišťují po přepnutí do vstupního režimu definovanou hodnotu logické hodnoty 0 nebo 1. Na vstupu hradla může také být Schmittův klopný obvod, který zajistí definované chování vstupu, při pomalých přechodech mezi logickými hodnotami 0 - 1 a zpět.

Vnitřní zapojení vždy významně ovlivňuje elektrické vlastnosti hradla. Proto jej výrobci společně s tabulkou elektrických parametrů zveřejňují. Lze pak snadno najít nejvhodnější zapojení.

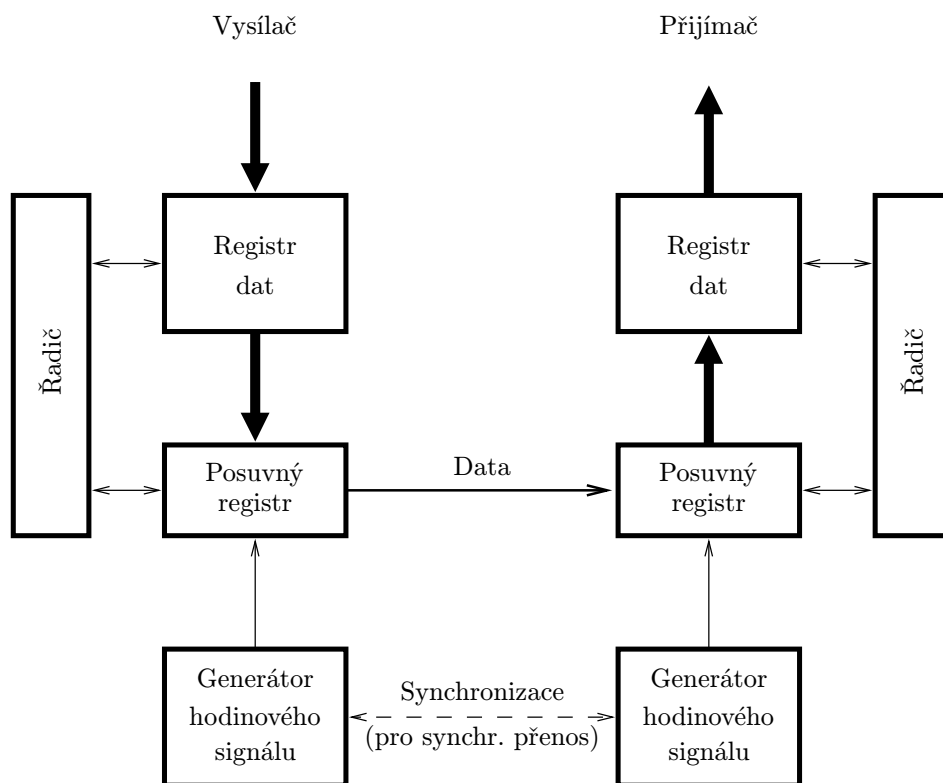
1.8.2 Sériové rozhraní

Sériový přenos dat je v praxi stále více používán. Dovoluje efektivním způsobem přenášet data na relativně velké vzdálenosti při použití minimálního počtu vodičů.

Hlavní nevýhodou sériového přenosu je nižší přenosová rychlost, tedy potřeba delšího časového intervalu pro přenesení potřebného množství informací. Další nevýhoda plyne z nutnosti vystřídat na jednom vodiči všechna data. Je tedy nutno data zakódovat a na přijímači správně dekodovat.

Protože je sériový přenos využíván v počítačové technice velmi často, přinesl vývoj řadu komunikačních standardů. Ty se liší počtem přenosových vodičů, velikostí napěťových úrovní pro logické hodnoty, rychlostí přenosu, zabezpečením a pod.

Podle vzdálenosti, na kterou jsou data přenášena, dělíme komunikaci na dvě základní kategorie:



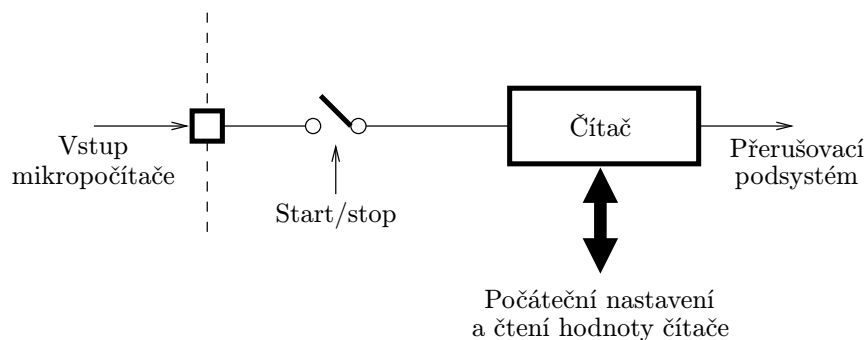
Obrázek 4: Sériový přenos

1. Komunikace mezi elektronickými zařízeními - jedná se o komunikaci na větší vzdálenosti, např. mezi řídicím počítačem a podřízenými stanicemi, nebo přenos dat z měřených míst technologického procesu. Jde o přenos synchronní nebo asynchronní, typicky pomocí RS232 nebo RS485.
2. Přenos dat uvnitř elektronického zařízení - používá se pro přenos dat mezi integrovanými obvody jedné aplikace. Typickým standardem je I²C pro komunikaci mezi integrovanými obvody.

Sériový synchronní a asynchronní přenos dat

Tento způsob komunikace patří mezi nejstarší a nejčastěji používané. Každý počítač je dnes vybaven sériovou linkou RS232, u které je možno snadno nastavit parametry přenosu, jako rychlost, parita, počet bitů ve slově a zabezpečení.

Napěťové hodnoty pro logickou hodnotu 0 a 1 jsou dané normou a pro konverzi na úroveň TTL se vyrábí speciální obvody.



Obrázek 5: Čítač vnějších událostí

Na obr. 4 je blokové schéma sériového přenosu dat jedním směrem. Při synchronním přenosu jsou generátory hodinového signálu vzájemně synchronizovány. Tím je zabezpečena správnost přenosu jednotlivých bitů. Nevýhodou je nutnost synchronizace generátorů, což obvykle vyžaduje další vodič, nebo pomocné obvody, které dokážou synchronizaci udržet rekonstrukcí dat z linky.

U asynchronního přenosu není nutná přesná synchronizace posuvných registrů. Tento princip dovoluje jisté malé odchylky u přijímače a vysílače. Sériová linka zůstává při nepřítomnosti dat v klidovém stavu v úrovni logické „1“. Přenos dat se zahajuje START bitem, který je vždy „0“. Přijímač musí detekovat sestupnou hranu na lince a od tohoto okamžiku zahajuje proces příjmu dat. Délka souvisle přenesených dat je pak omezena, aby se udržela vzájemná synchronizace v požadované toleranci.

Výhodou je větší volnost synchronizace, ale za cenu nižší přenosové rychlosti, protože s každými daty se musí navíc přenést minimálně START a STOP bit.

1.8.3 Čítače a časovače

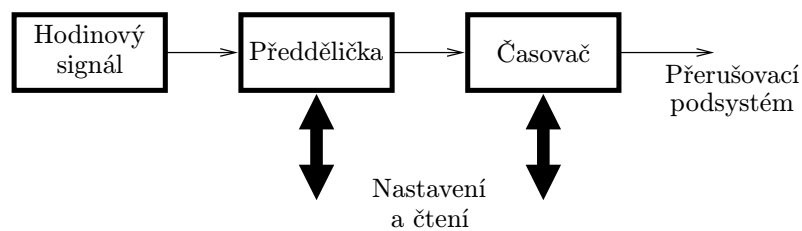
Do skupiny nejpoužívanějších periférií mikropočítače určitě patří čítače i časovače.

Čítač vnějších událostí

Čítač je N-bitový registr, který nejčastěji čítá vnější události. Obvykle lze nastavit, zda provádí inkrementaci při náběžné, nebo sestupné hraně vnějšího signálu. Při přetečení čítače se obvykle automaticky předává výzva do přerušovacího podsystému mikropočítače.

Programově se nastavuje počáteční hodnota čítače a také je možno čítač v libovolné chvíli odpojit a opět připojit k vnějšímu signálu. Příklad zapojení čítače je na obrázku 5.

Časovač



Obrázek 6: Časovač

Časovač se od čítače příliš neliší. Není ale inkrementován vnějším signálem, ale přímo vnitřním hodinovým signálem používaným pro řízení samotného mikropočítače. Lze tak úměrně přesnosti zdroje hodinového signálu zajistit řízení událostí a chování v reálném čase. Při přetečení časovače se i zde může automaticky předávat signál do přerušovacího podsystému mikropočítače.

Kromě nastavení počáteční hodnoty, lze programově obvykle ještě nastavit předděličku před samotným časovačem a tak lépe přizpůsobit chování časovače aplikaci, do které bude mikropočítač nasazen. Jednoduché schéma časovače je na obrázku 6

1.8.4 A/D převodníky

Fyzikální veličiny, které vstupují do mikropočítače, jsou většinou reprezentovány analogovou spojitou formou. Obvykle napětím, proudem, nebo odporem. Pro zpracování počítačem však potřebujeme informaci v digitální, tedy číselné formě. K tomuto účelu slouží analogově-číslíkové převodníky (analog/digital), kde se převádí analogová veličina, nejčastěji napětí, na číslo. Existuje několik základních typů těchto převodníků a ty nejčastěji používané si probereme.

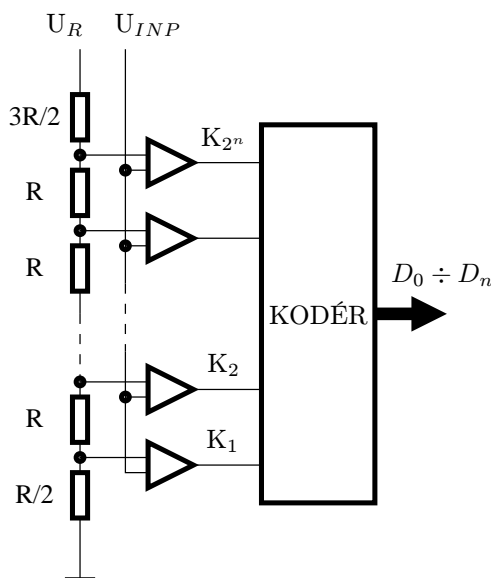
Komparační A/D převodníky

Jde o princip porovnávání měřené veličiny (napětí) s referenční hodnotou, rozdělenou na několik hodnot v určitém poměru, např. odporovou děličkou. Rozdělením referenčního napětí na několik hodnot dostáváme paralelní převodník. Takovéto převodníky jsou velmi rychlé a hodí se pro zpracování signálů.

S větším počtem komparátorů roste rozlišovací schopnost převodníku. Počet komparátorů ovšem pro lepší rozlišení narůstá velmi rychle. Následné možné rozdělení měření do několika kroků snižuje požadavek na počet komparátorů, ale zároveň prodlužuje dobu měření.

Příklad komparačního převodníku je na obr. 7.

A/D převodníky s D/A převodem



Obrázek 7: Komparační převodník

Základem těchto převodníků je použití jediného komparátoru a proměnného zdroje referenční hodnoty. Podle způsobu řízení referenční hodnoty rozlišujeme dva základní druhy převodníků: **sledovací** a **aproximační**.

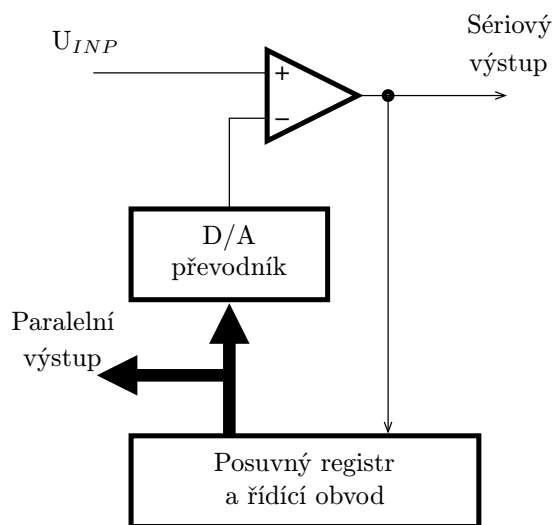
Převodníky sledovací mění vždy referenční hodnotu o jeden krok nahoru nebo dolů. Tento jednoduchý princip ale není vhodný pro skokově se měnící veličiny. Např. 8 bitový převodník by musel v krajním případě projít všech 256 možných referenčních hodnot. Proto se tyto převodníky hodí pro měření pomalu se měnících veličin, jako teplota či vlhkost.

Rychlejší převodník získáme při použití efektivnějšího algoritmu. Vhodná je např. aplikace známé metody půlení intervalu pro vyhledávání v setříděném seznamu. Takto získáme **aproximační** převodník. Referenční hodnota se na počátku nastaví na poloviční hodnotu mezi minimum a maximum. Podle výsledku komparátoru se pak hodnota posune o polovinu zbytku intervalu vždy nahoru, nebo dolů. Počet porovnávacích kroků je tedy maximálně roven počtu bitů rozlišovací schopnosti převodníku.

Na obr. 8 je jednoduché blokové schéma převodníku s postupnou aproximací. Ovládání posuvného registru lze snadno realizovat i programově.

Integrační A/D převodník

Na obr. 9 je schéma D/A převodníku s velkou rozlišovací schopností, obvykle 12 a více bitů. Integrátor integruje po pevně stanovenou dobu T_1 vstupní napětí U_{INP} . Po skončení této doby se přepne vstup P_1 integrátoru na napětí U_R opačné polaroty. Nyní se po dobu T_2 integruje referenčním napětím. Doba T_2 je závislá na napětí na konci periody T_1 . Délka T_2 je



Obrázek 8: Převodník s postupnou aproximací

tedy dána vztahem:

$$T_2 = \frac{U_{INP}}{U_R} \cdot T_1$$

Pokud je T_1 a U_R dáno, změříme čas T_2 a můžeme vypočítat:

$$U_{INP} = \frac{T_2}{T_1} \cdot U_R$$

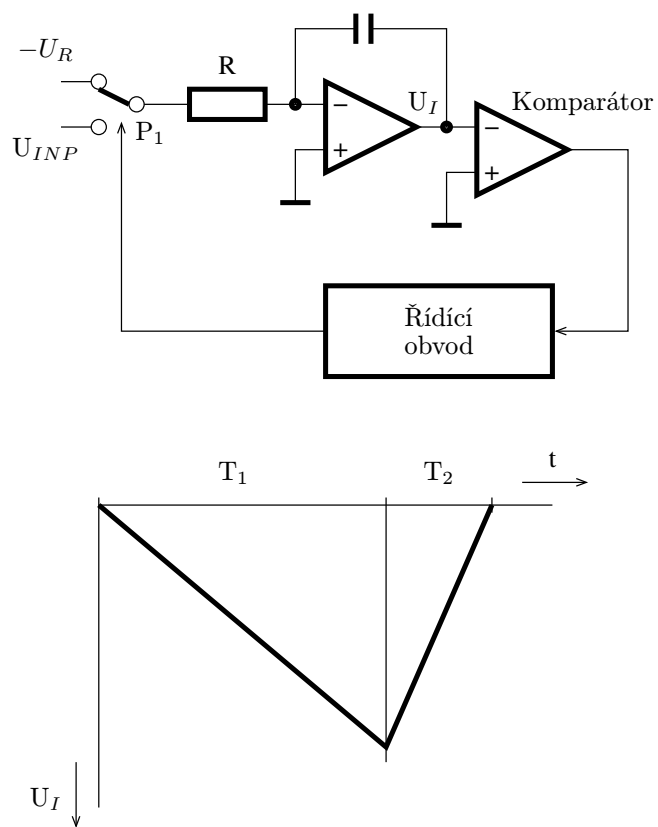
Tento princip nazýváme metodou dvojité integrace. Kondenzátor použitý v zapojení integračního převodníku ale nelze vyrobít jako součást integrovaného obvodu, proto se používá jako externí součástka.

Převodníky s časovacím RC článkem

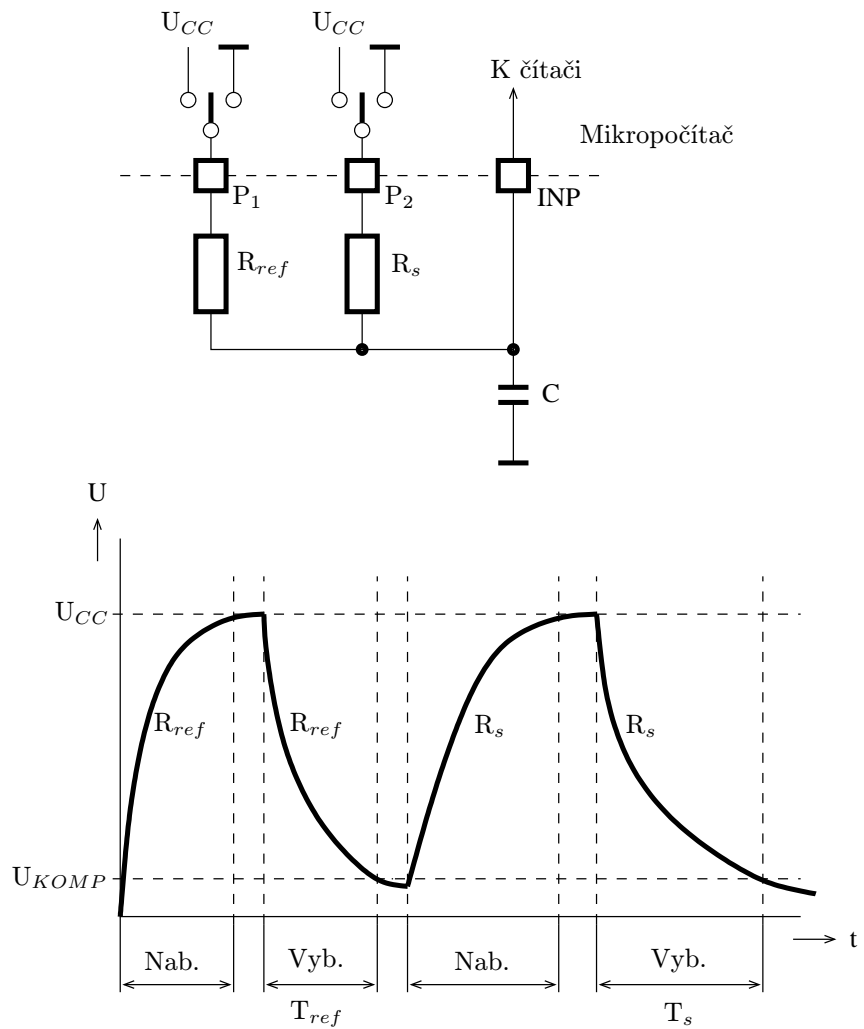
Ne vždy je analogová veličina reprezentována napětím či proudem. Často se různé veličiny měří pomocí snímačů, které mění svůj odpor v závislosti na dané veličině - např. tenzometry či termistory. Zde by převod na napětí znamenal jeden článek v A/D převodníku navíc.

Nejčastěji se využívá RC článků, kde doba nabití a vybití kondenzátoru je přímo úměrná hodnotě odporu i kapacity. Měření času pak lze snadno realizovat pomocí čítače, nebo časovače. Při měření je nutné srovnání s referenčním odporem. Porovnáváním dvou odporů eliminujeme vlastnosti kondenzátoru.

Na obr. 10 je zapojení snímače R_s . Jeho hodnotu získáme tak, že nejprve necháme nabíjet kondenzátor přes referenční odpor R_{ref} . Když napětí dosáhne hodnoty U_{CC} , spustíme časovač a necháme kondenzátor přes stejný



Obrázek 9: A/D převodník s dvojitou integrací



Obrázek 10: Převodník R/D s měřením doby vybíjení kondenzátoru

odpor vybijet. Při dosažení U_{KOMP} na vstupu INP se čítač zastaví. Získáme tak čas T_{ref} .

Stejný postup provedeme s použitím odporu R_s a získáme tak čas T_s . Ze dvou známých časů a hodnoty R_{ref} můžeme vypočítat:

$$R_s = R_{ref} \cdot \frac{T_s}{T_{ref}}$$

Tímto jednoduchým způsobem získáme převodníky s rozlišovací schopností 12 a více bitů.

1.8.5 D/A převodníky

Reálné prostředí, do kterého mikropočítače nasazujeme, vyžaduje nejen reakci na naměřené veličiny, ale i zásahy do tohoto prostředí. Tyto řídicí podněty musí být často také v analogové formě. K tomuto účelu používáme **číslicově-analogové převodníky** (digital/analog - D/A). Používáme dva základní principy: PWM a paralelní převodníky.

PWM (pulse-width modulation) - šířková modulace pulzů

Číslicový signál na výstupu mikropočítače má obvykle dvě konstantní napěťové úrovně, logickou „1“ při napětí U_1 a logickou „0“ při napětí U_0 . Pokud necháme na výstupu hodnotu „1“ po dobu T_1 a hodnotu „0“ po dobu T_0 , můžeme vypočítat střední hodnotu napětí:

$$U_{PWM} = U_0 + (U_1 - U_0) \cdot \frac{T_1}{T_0 + T_1}$$

kde součet času T_0 a T_1 označujeme T a je to časová perioda.

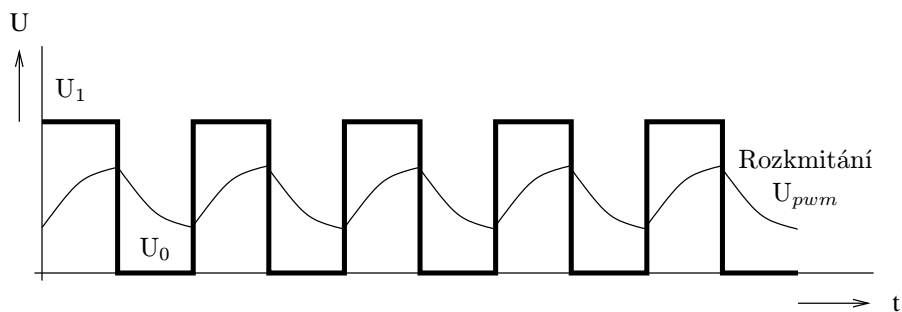
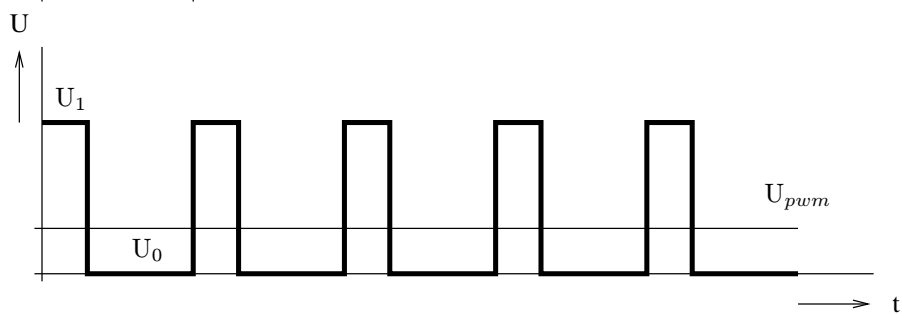
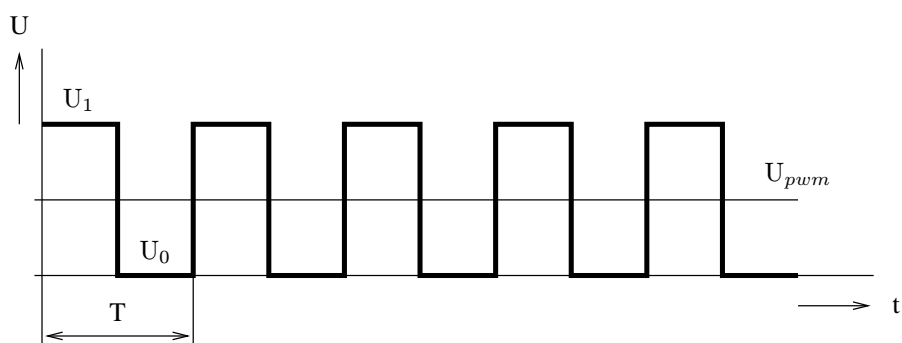
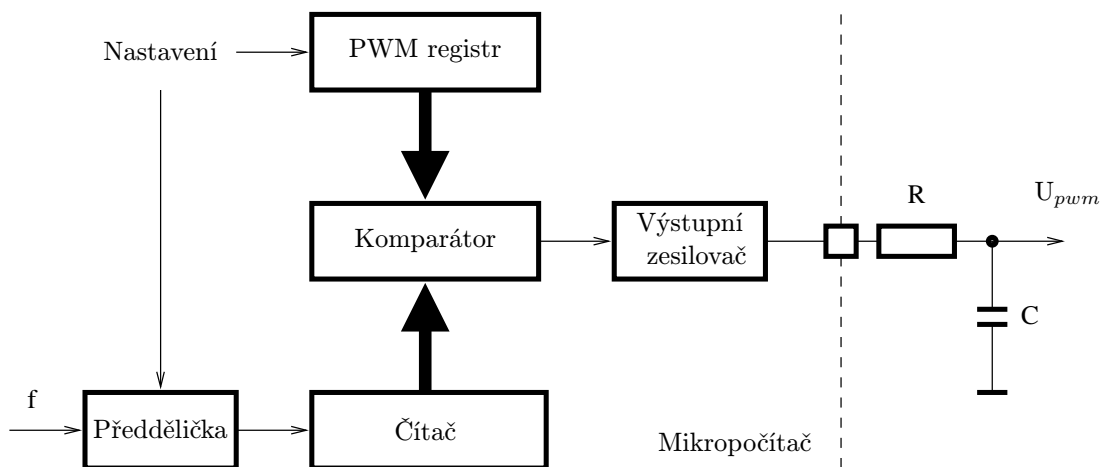
Příklad PWM převodníku a jeho chování je na obrázku 11. Pro převod šířkově modulovaných pulzů na analogovou veličinu slouží RC článek. Pro správnou činnost filtru je nutné, aby časová konstanta RC byla výrazně větší, než T . Musí tedy platit:

$$R \cdot C \gg T$$

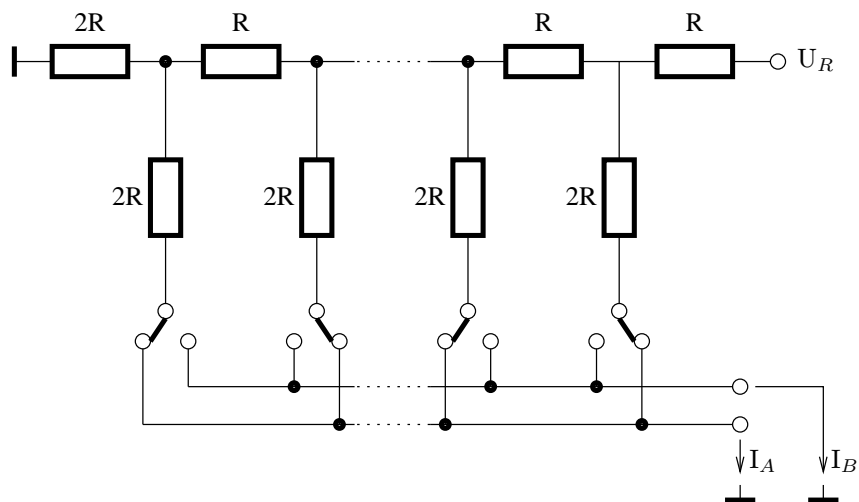
Nebude-li tato podmínka splněna, výstup převodníku bude nestabilní. Vnitřní odpor výstupu mikropočítače můžeme připočítat k RC článku, čímž časovou konstantu ještě prodloužíme. Musíme si ale uvědomit, že právě časová konstanta RC článku určuje poměrně velké zpoždění převodníku na požadovanou změnu výstupu.

Princip PWM lze v mikropočítači realizovat pomocí technických prostředků, nejčastěji pomocí čítače, nebo i programově. Počet bitů použitého čítače N určuje rozlišovací schopnost převodníku. Výstupní hodnota může mít 2^N různých úrovní.

Paralelní převodníky



Obrázek 11: Příklad PWM převodníku a jeho chování



Obrázek 12: Příklad R-2R převodníku s proudovými spínači

Základní nedostatek PWM převodníků, tedy velké zpoždění, odstraňují paralelní převodníky. Jsou založeny na přímém převodu číselné hodnoty na stejnosměrný proud.

Základem převodníku je většinou odporová síť, na níž se vytvářejí jednotlivé částečné proudy. Ta může být sestavena z odporů s **váhově řazenými hodnotami**. Výroba takové sady odporů je ale poměrně náročná, protože je nutno dodržet poměr mezi odpory 1:2:4:8:...:64:128....

Proto se používá síť **R-2R**. Používají se jen odpory hodnoty R a 2R. Na obrázku 12 je ukázka realizace takového D/A převodníku. Výstupem je proud I_a a komplementární proud I_b . Součástí převodníku musí být také zdroj referenčního napětí.

1.8.6 Obvody reálného času (RTC - Real Time Clock)

V mnoha aplikacích s použitím mikro počítačů je potřeba dodržovat přesnou časovou souvislost řízených událostí. Jde tedy o řízení v reálném čase. Ne vždy ale taková posloupnost dostačuje a je nutno pro potřebu řízení udržovat skutečný čas, tedy hodiny, minuty, sekundy a případně i zlomky sekund. Pro řadu aplikací pak navíc musíme udržovat nejen denní čas, ale i kalendář, tedy týdny, měsíce a roky. Pro tyto účely slouží obvody **hodin reálného času - RTC**.

Při použití obvodů RTC je obvykle nutno vyřešit dva základní problémy. Velká část aplikací pracuje s napájením ze sítě a při výpadku dochází ke ztrátám dat. Tedy i skutečného času. Proto je potřeba zajistit pro zálohování hodnot v RTC obvodu **záložní zdroj** pro udržení nepřetržité činnosti obvodu. Samozřejmě při návrhu obvodu je třeba uvažovat o velmi nízkém

odběru proudu, aby byla činnost RTC zajištěna po dostatečně dlouhou dobu z bateriového zdroje.

Druhým problémem je čtení dat z RTC obvodu. Čas je hodnota neustále se měnící. Např. pokud zahájíme čtení hodnoty v čase 10:59:59, může se snadno stát, že po přečtení prvních dvou hodnot, v našem případě hodin, se čas posune na 11:00:00, a čtení dalších hodnot bude neplatné. Proto se tyto problémy řeší buď technicky pomocnými registry v RTC obvodu, nebo vhodným programovým řešením.

1.8.7 Speciální periférie

Řada mikropočítačů je navrhována pro speciální účely, a tomu také odpovídají i jejich periférie:

- Řízení dobíjení baterií, a to nejen suchých typu NiCd, NiMH a Li-Ion, ale i baterií s elektrolytem.
- Pro telekomunikační techniku jsou některé mikropočítače vybaveny dvoutónovým multifrekvenčním generátorem a přijímačem.
- Pro TV přijímače obsahují obvody pro zobrazování informací „On Screen“.
- Vysílače a přijímače IR signálu, tedy modulaci, demodulaci a kódování.
- USB rozhraní typu klient.
- Řadiče LCD a LED zobrazovacích jednotek, které vyžadují dynamické řízení.