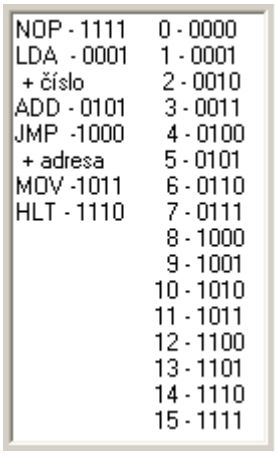
Model procesoru

Jedná se o blokové schéma složené z registrů, paměti RAM, programového čítače, instrukčního registru, sčítačky a řídicí jednotky, které jsou propojeny sběrnicemi. Tento model má dva stavy: *programování* a *provoz*. Přepnutí do stavu *programování* umožňuje zápis hodnot navolených na vstupu do paměti RAM pomocí tlačítek *krokování* a *zápis do RAM*. Po přepnutí do stavu *provoz* a spuštění programu tlačítkem *start/stop* lze sledovat činnost jednotlivých součástí mikroprocesoru a přesun dat, instrukcí a řídicích signálů po sběrnicích, a to s volitelnou rychlostí. Případně můžeme program odkrokovat pomocí tlačítka *ručně*.

Jako pomůcka jsou v pravém dolním rohu vypsány binární kódy čísel od 0 do 15 a binární kódy příkazů, které máme dispozici (obr.21).



Obr.21

NOP (no operation) – Příkaz pro „nicnedělání“. Používá se k přidání časového zpoždění k programu, pro rezervování místa v programu, pro pozdější přidání instrukce, nebo při odstranění instrukce aniž bychom přepisovali celý program.

LDA (load accumulator) – Příkaz pro načtení čísla do registru A. Po tomto příkazu následuje číslo v binárním kódu, které chceme načíst.

ADD (addition) – Příkaz pro sečtení obsahu registrů A a B. Výsledek se vkládá do registru A.

JMP (jump) – Příkaz, který nařizuje programovému čítači, aby skočil k požadované adrese v programové paměti. Po tomto příkazu následuje číslo v binárním kódu (0-15), označující požadovanou adresu v paměti RAM.

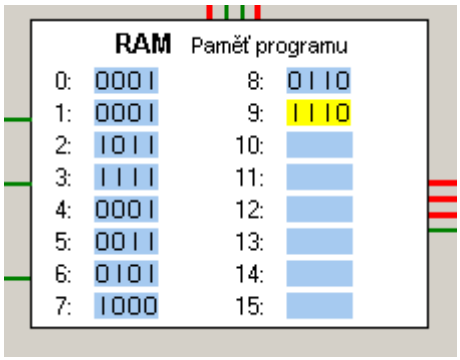
MOV (move) – Příkaz pro načtení obsahu registru A do registru B.

HLT (halt) – Příkaz, který vyřazuje z činnosti hodinové pulsy. Umisťuje se nakonec programu.

Jako příklad pro pochopení činnosti mikroprocesoru jsem zvolil program, který načte do registru číslo 3 a k tomuto číslu neustále přičítá jedničku.

- Mikroprocesor přepneme do stavu *programování*.
- Na první místo (řádek 0) v paměti RAM vložíme příkaz LDA pro načtení čísla do registru A: Tlačítky vedle přepínání stavů *programování- provoz*, nastavíme hodnotu 0001. Tlačítkem *zápis do RAM* se provede zapsání tohoto čísla do aktivního (žlutého) řádku paměti RAM.
- Tlačítkem *krokování* se postoupí na další řádek paměti.

- Dále následuje číslo 0001, což je číslo 1, které budeme přičítat.
- Toto číslo přesuneme z registru A do registru B příkazem MOV – 1011.
- Aby byla ukázána funkce všech příkazů, je v řádku 3 vložena instrukce NOP – 1111.
- Dále následují kódy 0001 a 0011, které představují načtení čísla 3 do registru A (LDA, 3).
- Příkaz ADD (0101) provede sečtení obsahu registrů a zapsání výsledku do registru A.
- Do dalších řádků paměti vložíme příkaz JMP – 1000 s adresou 0110, která značí, že se má provést skok na 6. řádek paměti. Na tomto řádku se nachází instrukce ADD, která opakovaně přičítá obsah registru B (číslo 1) k obsahu registru A.
- Na posledním řádku se nachází příkaz HLT – 1110 pro zastavení činnosti mikroprocesoru. Tato instrukce se však provádět nebude, protože se díky příkazu JMP bude neustále provádět přičítání jedničky k obsahu registru A.
- Zastavení činnosti se v tomto případě musí provést ručně. Na obr.22 je znázorněn program zapsaný v paměti RAM.



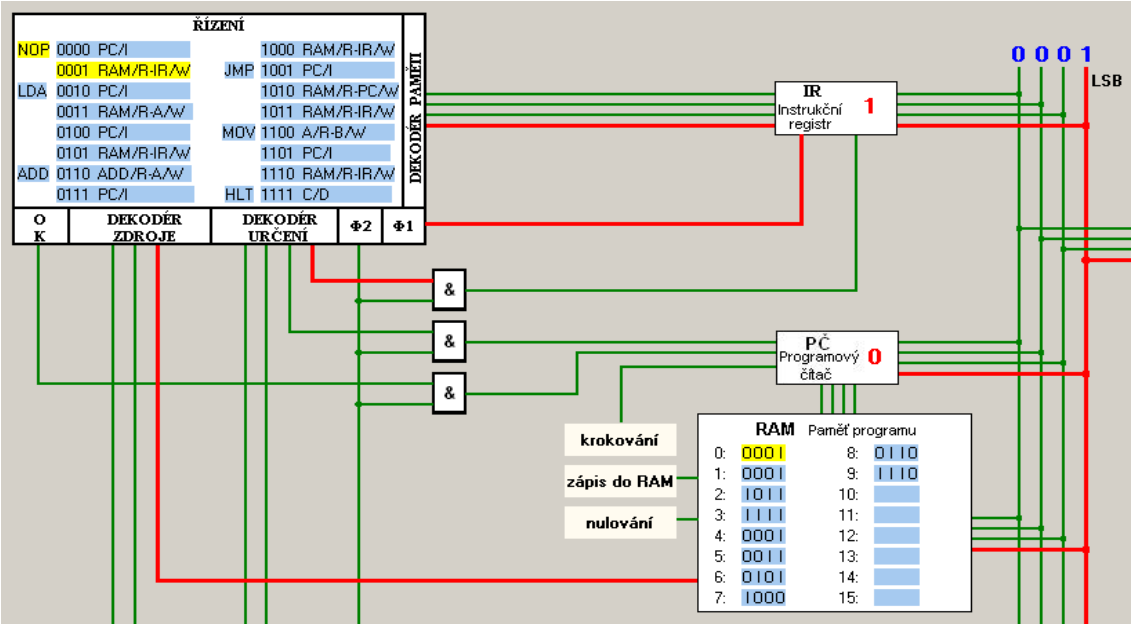
Obr.22

Tímto je programování dokončeno a můžeme přejít k vykonávání programu. Mikroprocesor přepneme do stavu *provoz*. Tlačítkem *start/stop* se mikroprocesor uvede do činnosti. Máme k dispozici sedm rychlostí vykonávání programu. Program se také může odkrokovat tlačítkem *ručně*.

Na začátku vykonávání programu je registr instrukcí i programový čítač vynulován. Registr instrukcí tudíž ukazuje na řádek 0000 v paměti ROM. Zde je uložena mikroinstrukce inkrementace programového čítače. Na tuto mikroinstrukci však procesor nereaguje, protože k tomu je třeba hodinový impuls Φ2 ve stavu *log1*.

Příchodem prvního hodinového pulsu Φ1 dojde ke zvýšení obsahu registru instrukcí o 1, tím je v paměti ROM aktivován řádek 0001, obsahující mikroinstrukci čtení z paměti RAM – zápis do instrukčního registru (obr.23). Z dekodéru zdroje a dekodéru určení jsou vyslány povely pro tyto operace.

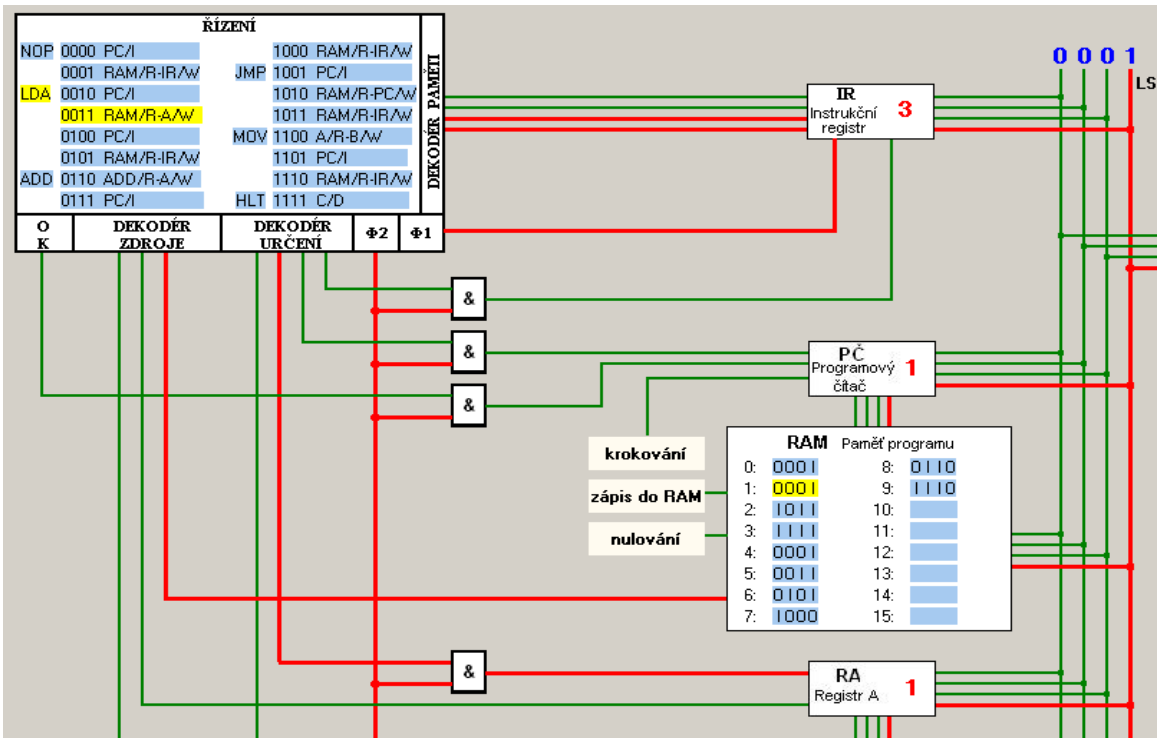
V paměti RAM je na nulté pozici umístěn příkaz LDA (0001). Toto číslo se ihned načte na sběrnici a s příchodem pulsu Φ2 (díky hradlu AND) se provede jeho zapsání do instrukčního registru. V tomto případě žádná změna nenastane, protože instrukční registr je na tuto hodnotu již nastaven.



Obr.23

S příchodem dalšího pulsu Φ1 se zvýší obsah instrukčního registru na hodnotu 2 a je aktivován řádek 0010 v paměti ROM, s mikroinstrukcí inkrementace programového čítače. S pulsem Φ2 se zvýší jeho obsah o 1, tím je aktivován následující řádek v paměti RAM.

Při dalším kroku se opět postoupí v paměti ROM o řádek dále na mikroinstrukci čtení z paměti RAM – zápis do registru A. Číslo 0001 z paměti RAM se zapíše na sběrnici a s příchodem pulsu Φ2 dojde k jeho zapsání do registru A (obr.24).

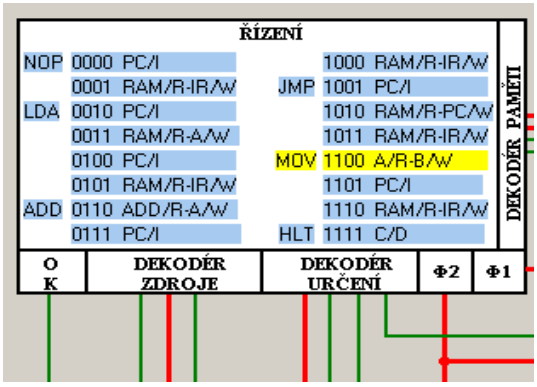


Obr.24

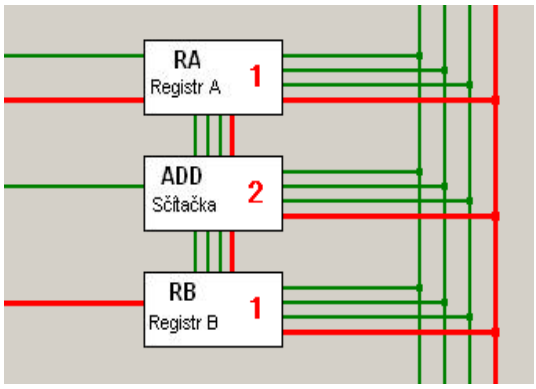
Pamětí ROM se pokračuje dále na mikroinstrukce 0100 – inkrementace programového čítače a 0101 – čtení z RAM – zápis do instrukčního registru. Tím skočí čítač programu na následující buňku paměti RAM.

Zde je zapsána další instrukce, tou je MOV (1011). Toto číslo se přepíše při Φ2 do instrukčního registru. 1011 je číslo řádku, který předchází prvnímu řádku s instrukcí MOV.

Registr instrukcí obsahuje vyrovnávací registr, který zasílá jeho obsah k řídicím obvodům až s náběžnou hranou Φ1, to už ale dojde k inkrementaci instrukčního registru, a tudíž aktivuje první mikroinstrukci instrukce MOV. Tou je mikroinstrukce čtení z registru A – zápis do registru B (obr.25, obr.26).



Obr.25



Obr.26

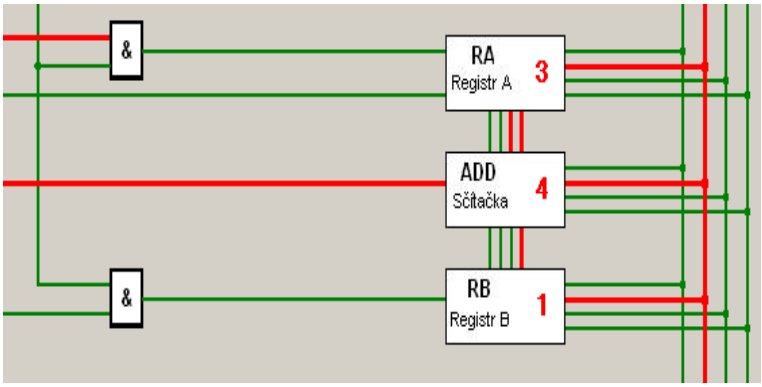
Po přepsání obsahu registru A do registru B následují opět mikroinstrukce inkrementace programového čítače a zapsání dalšího příkazu do instrukčního registru.

Tímto příkazem je NOP (1111), obsahující pouze dvě mikroinstrukce: inkrementaci programového čítače a načtení další instrukce z RAM do instrukčního registru.

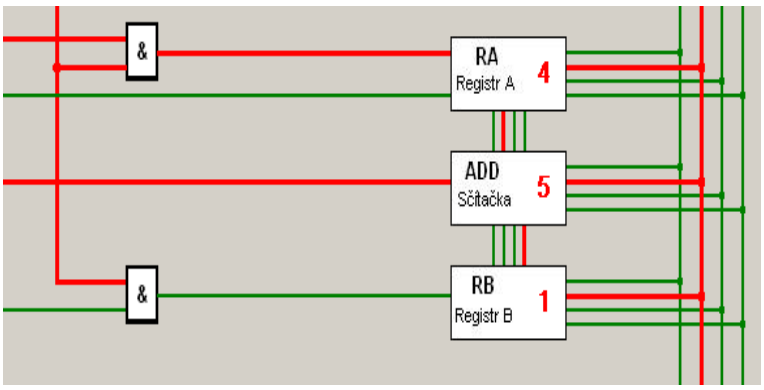
Následuje příkaz načtení čísla 3 do registru A (0001, 0011). Operace LDA byla popsána již dříve.

Nyní registr A obsahuje číslo 3 a registr B číslo 1, můžeme tudíž přistoupit k jejich sečtení. Výstupy registrů A a B jsou trvale připojeny ke sčítačce, která neustále provádí součet jejich obsahů, proto stačí přepsat obsah sčítačky do registru A.

K tomu slouží příkaz ADD (0101), jehož první mikroinstrukce je čtení ze sčítačky – zápis do registru A. Na obr.27 je vidět čtení ze sčítačky na sběrnici. Z řídicích obvodů přichází i povel pro zápis do registru A, ale čeká se na impuls Φ2 (obr.28).



Obr.27



Obr.28

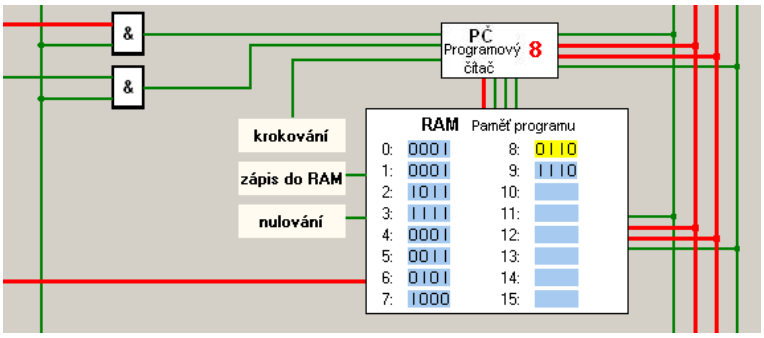
Opět následují mikroinstrukce inkrementace programového čítače a zápis dalšího příkazu do instrukčního registru.

Jelikož chceme přičítat k výsledku jedničku, stačí stále opakovat příkaz ADD. K tomu nám poslouží příkaz JMP (1000), umožňující skok na libovolnou buňku v paměti RAM.

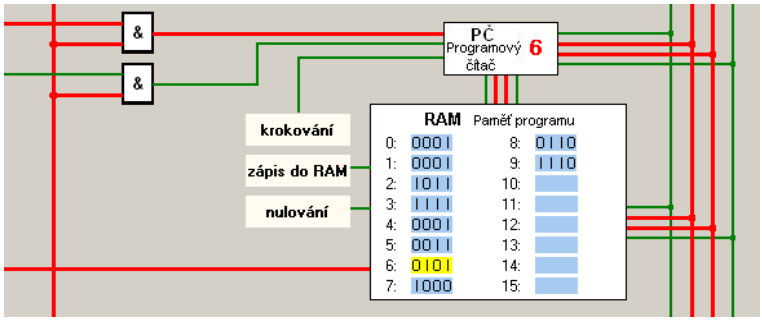
První mikroinstrukcí tohoto příkazu je inkrementace programového čítače, tím se v paměti RAM aktivuje následující řádek, v němž je uložena adresa, kterou má procesor pokračovat. My potřebujeme skočit zpět na příkaz ADD, který je umístěn na 6. řádku, a proto je zde uloženo číslo 0110.

Následuje mikroinstrukce čtení z paměti RAM – zápis do programového čítače. Na obr.29 je vidět čtení požadované adresy na sběrnici, s příchodem Φ2 se provede zápis této adresy do programového čítače a aktivování požadovaného řádku v paměti RAM (obr.30).

Poslední mikroinstrukcí je čtení z paměti RAM – zápis do instrukčního registru. Při této operaci dochází k načtení požadovaného příkazu do řídicích obvodů.



Obr.29



Obr.30

Poslední instrukcí, která nebyla zmíněna je HLT. Je to příkaz obsahující pouze jednu mikroinstrukci – zastavení činnosti dvoufázových hodin. Tento příkaz se umisťuje na konec programu. V tomto případě se však procesor až na konec programu nedostane díky příkazu JMP. Zastavení činnosti procesoru se zde musí provést ručně.