

2.) Tétel:

Bitmozgató műveletek, Neumann Ciklus(a.):

Bitenkénti léptetés

Léptető utasításokat használ több alapfunkció (soros-párhuzamos konverzió), és a szorzás művelete is.

Léptetés (SHIFT)

A bitek értékeit egy helyi értékkel balra vagy jobbra léptetjük. Jobbra léptetés esetén a balszélső bit értéke vagy nulla, vagy az előjelbittel megegyező lesz, a jobb oldali bit értéke egy átviteli bit lesz, amit a gép eltárol. Balra léptetés esetén mindig nulla lesz a jobb oldali bit értéke, a balról kilépő érték pedig átvitelként jelenik meg.

A fixpontos számoknál a balra léptetés kettővel szorzásnak, a jobbra léptetés kettővel osztásnak felel meg.

Forgatás (ROTATE)

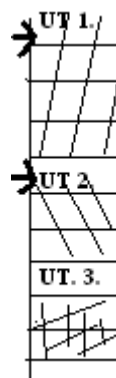
Balra rotálás esetén a kilépő bit vagy a legkisebb helyi értékre kerül, vagy eltárolódik átvitelként, és az előzőleg eltárolt átviteli bit kerül a legkisebb helyi értékre. Jobbra rotálás esetén a kilépő bit vagy a legmagasabb helyi értékre kerül, vagy eltárolódik átvitelként, és az előzőleg eltárolt átviteli bit kerül a legnagyobb helyi értékre.

Neumann ciklus: Egyetlen egy gépi utasítás ciklust ír le.

1. lépés: Utasítás betöltése a memória helyről. Utasítás regiszter betöltése IP-ről. *IP*: Utasítás mutató. Programutasítások, amelyet a számítógépnek végre kell hajtani.
2. lépés: Következő utasítás helyének meghatározása. IP beállítása, hogy a következő utasításra mutasson.
3. lépés: Utasítás végrehajtása.

Ezután visszaugrunk az elejére. Neumannnál volt még 1 feltétel, de már nincs. A processzor soha nem áll le. Ha nem dolgozunk akkor sem áll le.

Ábra: az utasítás kódjából derül ki, hogy hány bájtos az utasítás. Lehet hogy pl az első 4 bájtos, 2. 3 , 3. 2 stb...



CASE szelekciót felismerő automata(b.) : ????

Az LL(k) grammatika(c.)

Az $S \Rightarrow^* wx$ legbaloldalibb levezetés építése során eljutunk a $S \Rightarrow^* wA\beta$ mondatformáig és az $A\beta \Rightarrow^* x$ -et szeretnénk elérni, akkor az A-ra alkalmazható A > a helyettesítést egyértelműen meghatározhatjuk az x első k db szimbólumának előre olvasásával (ekkor és csak ekkor LL(k) nyelvtanról beszélhetünk).

Legyen $FIRST_k(\alpha)$ az α -ból levezethető szimbólumsorozatok k hosszúságú kezdő terminális sorozatainak halmaza, azaz $FIRST_k(\alpha) = \{x \mid \alpha \Rightarrow^* x\beta \text{ és } |x| = k\} \cup \{x \mid \alpha \Rightarrow^* x \text{ és } |x| < k\}$. Tehát a $FIRST_k(\alpha)$ az x első k darab szimbólumát, $|x| < k$ esetén pedig a teljes x-et tartalmazza. Ha $\alpha \Rightarrow^* \varepsilon$, akkor természetesen $\varepsilon \in FIRST_k(\alpha)$.

A G grammatika LL(k) grammatika, ha tetszőleges $S \Rightarrow^* wA\beta \Rightarrow w\alpha_1\beta \Rightarrow^* wx$ és $S \Rightarrow^* wA\beta \Rightarrow w\alpha_2\beta \Rightarrow^* wx$ levezetés párra $FIRST_k(x) = FIRST_k(y)$ esetén $\alpha_1 = \alpha_2$. Eszerint tehát ha egy grammatika LL(k) grammatika, akkor a már elemzett w utáni k darab terminális szimbólum az A-ra alkalmazható helyettesítési szabályt egyértelműen meghatározza.