

3.) Tétel:

Feltétel nélküli vezérlésátadás, processzor utasításrendszere, operandusok , címzési módok(a.):

Feltétel nélküli vezérlésátadás: Feltétel nélküli ugrásnál az utasításban szereplő címmel tölti fel a processzor az utasításslámláló regiszter tartalmát, amely a következó utasítás címe lesz és a program innen folytatódik. Assembly-ben: JMP címke

- nem elegáns használni
 - több nyelvben már nem is létezik ez az utasítás
 - vagy csupán végtelen ciklusból való kiugrásra lehetséges a használata
- a probléma vele többrétű
 - átláthatatlanná teheti a programstruktúrát , hibakeresést megnehezíti
 - a fordítóprogram nem tudja miatta jól meghatározni a program erőforrás- vagy memóriaigényét
- kivételkezelés programnyelvi támogatása kiválthatja a szükségességét
- FORTRAN0 vezette be, COBOL támogatja
- Pascalban címkére ugorhat blokkon belül, C-ben címkére ugorhat függvényen belül
- Modula-3-ban nincs GOTO (de van RETURN visszatérni eljárásokból, függvényekből), Java-ban nincs GOTO

A processzor utasításrendszere

Minden számítógépnek van egy belső memóriája az éppen futó programok és az adatok tárolására. A PC processzorok memóriája byte szervezésű, azaz minden byte-nyi memóriának van egy memória címe. Bármely két szomszédos byte egy 16 bites szót alkot. A gépi kódú utasítások négy részből állnak, méretük függ az utasítástól és a címzési módtól. Az általános utasítás szerkezet:

Prefixum|Operáció kód|Címzési mód|Operandus

1. A prefixum módosítja az utasítás értelmezését, pl. ezzel írható elő az ismétlések száma, vagy a megszakításkérés tiltása. Használata nem kötelező.
2. Az operáció kód adja meg, hogy a processzornak milyen műveletet kell végrehajtania. Használata kötelező.
3. A címzési mód az operandusok értelmezését adja meg. Nem minden utasításban található meg.
4. Az operandus lehet konstans, cím vagy címzéshez használt érték. Hossza változó, használata nem kötelező.

Az assembly utasítások pontosan megfelelnek egy gépi kódú utasításnak, azaz minden assembly utasításból egy gépi kód lesz.

Operandusok

Az Intel processzorokban egy utasítással csak egy memóriahely címezhető meg. Azaz, ha egy utasítás két operandusú, akkor csak az egyik lehet memóriacím, a másik regiszter kell, hogy legyen.

Regiszteroperandus

Az utasítás paraméterei regiszterek, amelyekben a művelethez szükséges adatok vannak. Az utasítás hossza a regisztermérettől függ.

Példa: MOV AX,BX

Közvetlen operandus

Az operandus az utasítás kódját tartalmazza. Példa: MOV AX,2

Címzési módok:

Direkt memóriacímzés

Itt az operandus egy, a Data szegmensben előre deklarált változó.

.DATA

adat DB "A"

.CODE

MOV AX, adat

Indirekt memóriacímzés

Itt az operandus az adat címét tartalmazza, nem az értékét.

Regiszter indirekt címzés

Példa: MOV AX,[BX] ;AX-be tölti a BX által megcímezett memória tartalmát

Indexelt, bázis-relatív címzés

.DATA

adat DB 10h,20h

.CODE

MOV AX,adat[BX] ;AX-be tölti az adat+BX címen lévő adatot

<code>

Bázis plusz index címzés

Példa: MOV AX,[BX][DI] ;AX-be tölti a BX+DI címen lévő adatot

Bázis plusz relatív címzés

Példa: MOV AX,adat[BX][DI] ;AX-be tölti az adat+BX+DI címen lévő adatot

Ez a megoldás használható többdimenziós tömbök kezelésére.

A stack címzése

Erre a bázisregiszteres címzési mód használható. A szegmenscímet az SS regiszter, a báziscímet a BP regiszter tartalmazza.

<code>

PUSH BP ;BP eredeti tartalmának mentése

MOV BP,SP ;A stack tetejének címe a BP-be

MOV AX,2[BP] ;Az első paraméter betöltése AX-be

MOV BX,4[BP] ;A második paraméter betöltése BX-be

POP BP ;BP eredeti tartalmának visszaállítása

A programterület címzése

IP-relatív címzés

Példa: JMP SHORT cím ;Közeli ugrás a címre A feltételes vezérlőátadó utasítások mindig IP-relatív címzéssel működnek.

Direkt címzés

Példa: JMP cím ;Direkt ugrás a szegmensen belüli címre

Indirekt címzés

Példa: JMP BX ;Ugrás a BX által címzett utasításra

IF szerkezetet felismerő automata(b.):

Determinisztikus véges automaták(c.):

Véges automaták

Az automata egy olyan konstrukció, mely egy input szóról képes eldönteni, hogy az helyes-e vagy sem.

1. Létezik egy input szalagja, amely cellákra van osztva, és minden cellában egy-egy karakter van. Az input szalag általában véges és olyan hosszú, mint az input szó.
2. Az input szalaghoz egy olvasófej tartozik, amely mindig egy cella fölött áll és ezen cellából képes kiolvasni a karaktert. Az olvasófej lépked a szalag cellái között egyesével, balra-jobbra.
3. Létezik egy output szalagja, ami cellákra van osztva, és minden cellában egy-egy karakter lehet. Az output szalag lehet véges vagy végtelen. Azokban a cellákban amelyekbe még nem írt semmit az automata, egy speciális karakter, a BLANK jel áll.
4. Az output szalaghoz egy író-olvasó fej tartozik. Ennek segítségével az automata egy jelet írhat vagy olvashat az aktuális cellából. Az író-olvasó fej léptethető a cellák fölött balra-jobbra.

5. Az automatának belső állapotai vannak, melyek között az automata lépkedhet egy megadott séma alapján. Az automata mindig pontosan egy állapotban van (aktuális állapot).

A legegyszerűbb automata-csoport a véges automaták. Nincs output szalagjuk, sem író-olvasó fej. Az input szalag hossza véges, épp olyan hosszú, mint az input szó (ugyanannyi cellából áll, amennyi a szó hossza).

Egy $M(Q, \Sigma, \Delta, q_0, F)$ ötöst véges automatának nevezzük, ahol

1. Q az automata állapotainak véges halmaza
2. Σ az elemzendő jelsorozat
3. Δ az automata mozgási szabályainak véges halmaza
4. q_0 az induló állapot, $q_0 \in Q$
5. F az elfogadó állapotok halmaza, $F \subset Q$

A véges automaták egy irányított gráffal szemléltethetők, és az egyes csomópontjai felelnek meg az automata állapotainak. Amennyiben egy karakter beolvasásának hatására az automata egyik állapotból egy másik állapotba megy át, akkor a két állapotnak megfelelő csomópontokat egy, az eredeti állapotból az új állapotba mutató, és az olvasott karaktert mint nevet (színt) viselő éllel kötjük össze.

Mozgási szabályok

A mozgási szabályok mondják meg, hogy egy adott állapotban (A), egy adott karakter (a) beolvasásának hatására milyen új állapotot vesz fel az automata: $\Delta(A, a) = B$. A mozgási szabályok összessége tulajdonképpen egy leképezés, amely az automataállapotok és az alfabéta karaktereinek direkt szorzatából álló halmazt képezi le az automataállapotok halmazára: $Q \times \Sigma \Rightarrow Q$.

Amennyiben minden állapot–karakter párhoz legfeljebb egy mozgási szabály tartozik, akkor az automata működése egyértelműen meghatározott, az automata determinisztikus. Ha minden állapot–karakter pár esetében van mozgási szabály, akkor az automata teljesen specifikált. Mindannyiszor, amikor egy mozgási szabályt alkalmazva egy karaktert figyelembe vettünk (beolvastunk), a jelsorozat következő karaktere lesz érvényes. Az automata az elemzett jelsorozatot elfogadhatja, vagy visszautasíthatja. Az elfogadásnak két, egyidejűleg teljesítendő feltétele van:

1. az automatának a szöveget végig kell olvasnia (minden állapot-karakter páros esetében létezik mozgási szabály)
2. az utolsó karakter elolvasása után az automatának elfogadó (vég) állapotba kell kerülnie

A visszautasítás csak nem teljesen specifikált automaták esetében fordulhat elő.