

LPJ - Labda Pattogtató Játék

(dokumentáció)

1. - A játékról

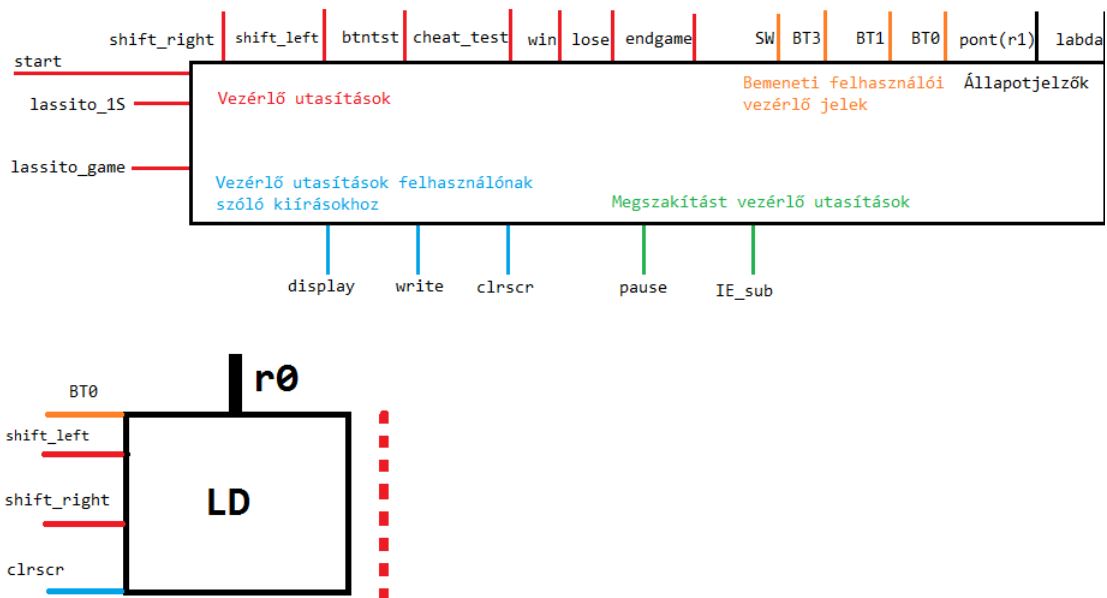
Játékszabály: a játékos 7 labdával kezd, amelyeket vissza kell pattintani a szemközti falra. Visszapattintani az FPGA kártyán található gomb használatával lehet akkor és csak akkor, mikor az elérte a pálya szélét. A játék a sikeres pattintást ponttal jutalmazza, a sikertelen próbálkozásért viszont labdavesztés jár. A játék akkor ér véget, ha a játékos elérte a 9 pontot, vagy elfogytak a rendelkezésre álló labdák.

A játék gyakorlati megvalósítása az FPGA kártyán a következőképpen néz ki: a labda helyzete a LED soron látható, míg a pontok a hétszegmenses kijelzőn követhetőek nyomon. A rendelkezésre álló labdák száma a szemléletes 5x7-es pontmátrix kijelző egy oszlopán látszik. További feliratok (PAUSE, SCORE) megjelenítéséhez a pontmátrix és a hétszegmenses kijelző együttes használatára volt szükség.

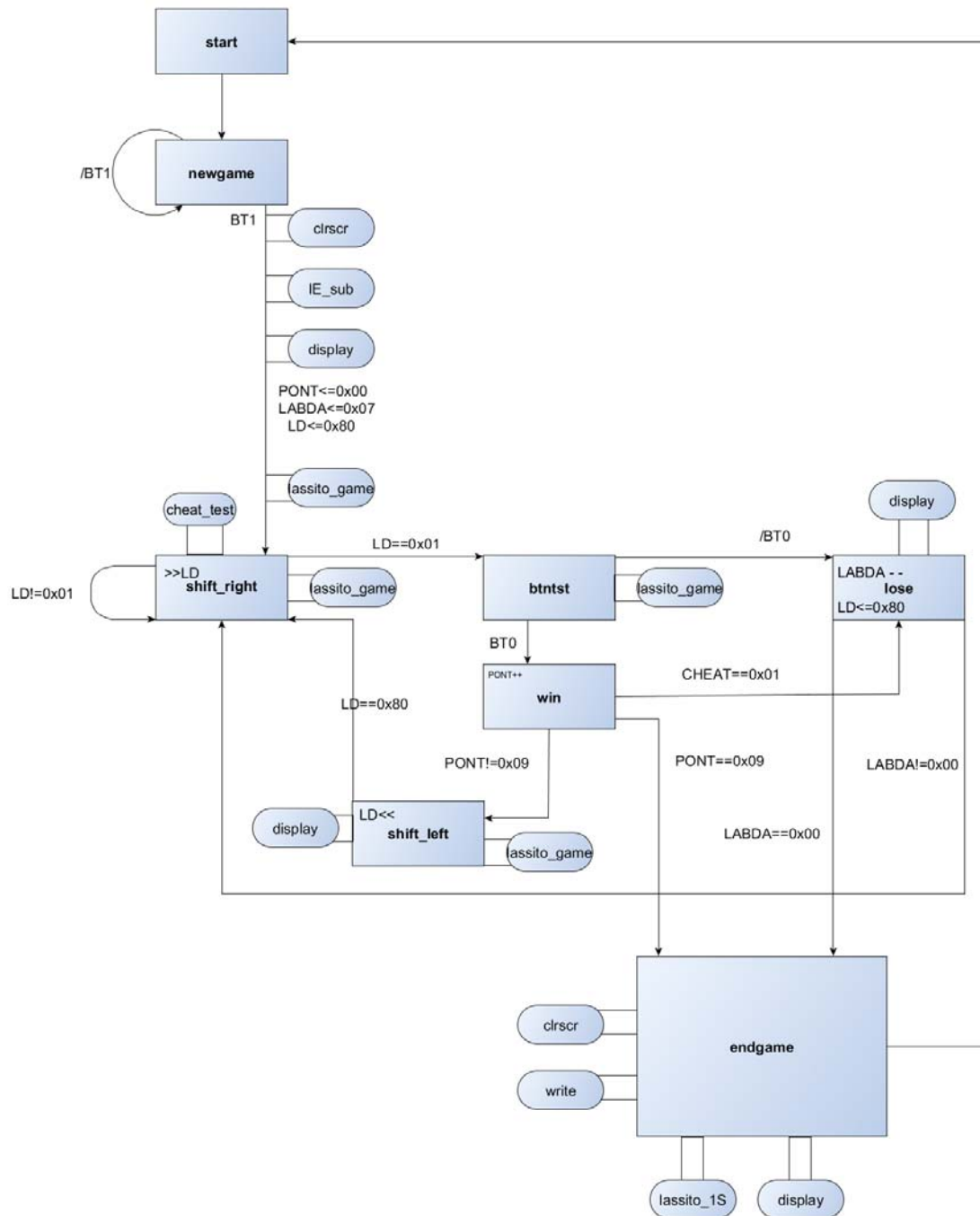
A vezérléshez a gombokat ill. kapcsolókat lehet használni, ezek szerepe a következő: BT0-pattintás, BT1-start, BT3-pause, valamint a SW kapcsolókkal a játék sebességét lehet befolyásolni, ehhez viszont ajánlott a bináris számrendszer ismerete.

2. - Megvalósítás

A program assemblyben íródott, a Minirisc processzor használatával. Az alábbiakban csak az assembly kódot részletezzük.



A program blokkvázlata fent látható. Az állapotdiagram - már amennyiben assembly nyelvből lehetséges ilyen készíteni - lent látható. A nagy szögletes dobozok a programrészeket, a kis kerekített téglalapok a különböző szubrutinokat jelzik.



Ez a diagram természetesen csak a programkód vázlatos működésének megjelenítésére alkalmas, a MINIRISC assembly nyelv jellegzetességeiből adódó regiszterhasználatot egyszerűsítve tünteti fel, az adatokat közvetlenül a perifériára helyezi.

A játék elején a program a felhasználóra vár, hogy az megnyomja a BT1-es gombot, és kezdődhessen a játék. Innentől kezdve a program a labda regiszterébe 0x80-t tölt, és azt

shifteli jobbra 0 beléptetésével. Innentől kezdve amúgy működik a megszakítás, amely a BT3-as gomb megnyomásával lép életbe. Ekkor a program egy PAUSE felirattal vár addig, míg a felhasználó le nem nyomja a BT1-es gombot. Ekkor a játék folytatódik, mintha mi sem történt volna. A játék minden egyes jobbra tolásnál ellenőrzi a BT0 gomb lenyomását a *cheat test* szubrutinnal. Ez azt vizsgálja, nem a „csal” a játékos a gomb folyamatos nyomvatartásával. Erre igazából csak az utolsó előtti pillanatban lenne szükség, hiszen a szubrutin a teszt eredményét jelző regisztert minden egyes alkalommal „kinullázza”. Ha a játékos nem pontosan a „labdaregiszter==0x01” pillanatban nyomja le a BT0 gombot - vagyis mikor a labda elért a pálya másik szélére - akkor semmiképpen sem nyerhet. Viszont ha ez sikerült, akkor a labda visszapattan, ami a regiszterben lévő érték balra shiftelésének feleltethető meg, szintén 0 beléptetésével. A játékos pontjai ekkor növekednek, hiszen a program a visszapattintás előtt megjárja a *win* programrészt, s ez a hétszegmenses kijelzőn is nyomon követhető. A *lose* programrész értelemszerűen azokat a teendőket végzi, amelyek a sikertelen próbálkozás után kell elvégezni: a pontok és a rendelkezésre álló „életek” (labdák) „dekrementálását”. A grafikus megjelenést a hétszegmenses és pontmátrix kijelzőn a program adatszegmens segítségével valósítja meg, itt tárolja a különböző számok „kódját”, valamint a pontmátrix felhasznált oszlopára írandó értékeket. A kiolvasásról és megjelenítésről két pointerregiszter gondoskodik, ennek részletezése a forráskódban található. LD==0x80 esetén a program előlről kezdi a jobbra shiftelést. A játék a *win* és a *lose* programrészekben ellenőrzi a pontok és az életek állását, és a diagramon jelzett feltételek szerint ugrik az *endgame* részre. Innentől kezdve a program kiírásokat végez különböző szubrutinok meghívásával (a diagramon csak a szubrutinok fajtái láthatóak, ennél a programrésznél igazából többször is meghívjuk ezeket). A *clrscr* a kijelző nullázását végzi, a *write* a SCORE feliratot, míg a *lassito 1S* az egy másodperces szoftveres lassítást végzi egy 24 bites számlálóval. A *display* meg a többször használt szubrutin a pontok megjelenítésére. A játék lassításáról (*lassito game*) még annyit, hogy ez is egy szoftveres 24 bites számláló, amely a kezdőértéket a kapcsolókról olvassa be, így lehetséges a játék nehézségének szabályozása (12 másodperces kezdőérték jelenti a kb. egy másodperces lassítást). A *lassito game* a diagramon jelzett helyeken hívódik meg.

A játék vége után, a pontok 3-szori felvillantását követően a program a start részre ugrik, és újra lehet próbálkozni.

A program forráskódja (megjegyzésekkel):

```

DEF LD      0x80          ; LED adatregiszter (írható/olvasható)
DEF BT      0x84          ; Nyomógomb adatregiszter (csak olvasható)
DEF BTIE    0x85          ; Nyomógomb megszakítás eng. regiszter (írható/olvasható)
DEF BTIF    0x86          ; Nyomógomb megszakítás flag regiszter (olvasható és a bit 1 beírásával töröl)
DEF DIG0    0x90          ; Kijelző DIG0 adatregiszter (írható/olvasható)
DEF DIG1    0x91          ; Kijelző DIG1 adatregiszter (írható/olvasható)
DEF DIG2    0x92          ; Kijelző DIG2 adatregiszter (írható/olvasható)
DEF DIG3    0x93          ; Kijelző DIG3 adatregiszter (írható/olvasható)
DEF COL0    0x94          ; Kijelző COL0 adatregiszter (írható/olvasható)
DEF COL1    0x95          ; Kijelző COL1 adatregiszter (írható/olvasható)
DEF COL2    0x96          ; Kijelző COL2 adatregiszter (írható/olvasható)
DEF COL3    0x97          ; Kijelző COL3 adatregiszter (írható/olvasható)
DEF COL4    0x98          ; Kijelző COL4 adatregiszter (írható/olvasható)
DEF SW      0x81          ; Kapcsoló adatregiszter

DATA
POINTS:
    DB 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F
;a pontok hétszegmenses "kódjai"
LIVES:
    DB 0x7F, 0x3F, 0x1F, 0x0F, 0x07, 0x03, 0x01, 0x00, 0x00
;a pontmátrix kódja (ez megoldható lenne egy egymásból való kiszámolással is)

CODE

start:          ;a program a start-ra ugrik
jmp newgame     ;a program a newgame-re ugrik
jmp pause       ;megszakításra ugorjon a pause interruptra

newgame:        ;itt indul a játék
mov r0, BT      ;"előtét loop"
and r0, #0x02   ;várunk a BT1 lenyomására
jz newgame      ;ha ez nem történik meg, akkor maradunk a hurokban

jsr clrscr      ;a clrscr szubrutin letakarítja a képernyőt

jsr IE_sub      ;az IE_sub beállítja a megszakítás engedélyezését a BT3-as gombra

sti            ;start interrupt
mov r8, #POINTS ;az r8-as regisztert a pontokra...
mov r9, #LIVES  ;...az r9-es regisztert az életekre állítjuk (adatpointerek)
jsr display     ;a display a státusz kivillantását intézi a továbbiakban
mov r1, #0      ;kezdetben 0 pont
mov r2, #7      ;kezdetben 7 labda
mov r0, #0x80   ;kezdetben a labda a bal szélén van
mov LD, r0      ;ez a ledre villantást jelzi a továbbiakban
jsr lassito_game ;ez meg a lassító loopot

shift_right:    ;játsszunk!
sr0 r0          ;jobbra shiftelünk 0 beléptetésével
jsr lassito_game
mov LD, r0
cmp r0, #0x01   ;elért a jobb szélére?
jz btntst       ;ha igen, a nyomógombot ellenőrizzük
jsr cheat_test  ;ha nem, csalást ellenőrizzük
jmp shift_right ;amúgy minden a régi kerékvágásban folytatódik

shift_left:     ;balra shiftelünk 0 beléptetésével
sl0 r0
jsr lassito_game
mov LD, r0
cmp r0, #0x80   ;megnézzük elért e a bal szélére?
jz shift_right  ;ha igen visszapattn
jmp shift_left  ;ha nem hát nem, folytatjuk a shiftelést

btntst:
mov r3, BT      ;Buttont bemozgatjuk az r3-as regiszterbe későbbi vizsgálódás céljából
tst r3, #0x01   ;le van nyomva a BT0 gomb?
jnz win         ;ha igen, nyertünk (hacsak nem csaltunk, de ez majd kiderül)
jmp lose        ;ha nem, veszítettünk

```

```

win:
    cmp r10, #1                ;megnézzük nem-e csalt a játékos (a gomb folyamatos nyomvatartásával)
    jz lose                    ;akkor bizony veszített!
    add r1, #1                  ;amúgy a pontot növeljük
    add r8, #1                  ;a pont-pointert léptetjük
    jsr display
    cmp r1, #9                  ;megnézzük, nem e 9 a pont?
    jz endgame                  ;ha igen, vége a játéknak
    jmp shift_left              ;amúgy visszapattan a labda

lose:
    add r9, #1                  ;a labda-pointer növelése
    sub r2, #1                  ;labdák számát csökkenteni kell
    jz endgame                  ;ha nulla, veszítettünk
    jsr display
    mov r0, #0x80                ;új labda érkezik balszélről...
    mov LD, r0
    jsr lassito_game
    jmp shift_right              ;...és megy jobbra

endgame:
    jsr clrscr                  ;üres kijelző
    jsr write                    ;SCORE
    jsr lassito_1S               ;1 mp
    jsr lassito_1S               ;1 mp

    jsr clrscr                  ;üres kijelző
    jsr display                  ;pontok kiírása
    jsr lassito_1S               ;háromszori villanás
    jsr clrscr
    jsr lassito_1S

    jsr display
    jsr lassito_1S
    jsr clrscr
    jsr lassito_1S

    jsr display
    jsr lassito_1S
    jsr lassito_1S

    jmp start                    ;a start programrészre ugrunk

cheat_test:
    mov r10, #0                ;Aki folyamatosan lenyomva tartja a gombot, az nem nyerhet!
    mov r3, BT                  ;a "cheatregiszter" kinullázódik
    tst r3, #0x01                ;Buttont bemozgatjuk az r3-as regiszterbe későbbi vizsgálódás céljából
    jz endtest                  ;le van nyomva a BT0 gomb?
    mov r10, #1                  ;ha nem, akkor vége is a tesztnek
    endtest:                    ;ha viszont igen, akkor a cheatregiszter egyest kap

rts

display:
    mov r7, (r9)                ;a pontmátrix kijelző 0. oszlopán
    mov COL0, r7                ;a játékos labdái
    mov r7, (r8)                ;míg a hétszegmenses kijelző 4.digitjén
    mov DIG3, r7                ;a játékos pontjai láthatók
    mov r7, #0x40                ;-
    mov DIG2, r7
    mov r7, #0xF3                ;P.
    mov DIG1, r7                ;x-P.

rts

```

```

lassito_1S:                ;szoftveres időzítés, kb 1 másodperc
    mov r13, #0            ; a 24 bites számláló beállítása
    mov r14, #0
    mov r15, #0
    loop_arena_1S:        ;a hurok
        add r13, #12
        adc r14, #0
        adc r15, #0
    jnc loop_arena_1S
rts

lassito_game:              ;a játék lassítója
    mov r13, #0            ;a SW perifériával
    mov r14, #0            ;tetszőlegesen állítható
    mov r15, #0
    mov r12, SW
    loop_arena_game:
        add r13, r12
        adc r14, #0
        adc r15, #0
    jnc loop_arena_game
rts

clrscr:                    ;a kijelző letisztítása
    mov r7, #0x00
    mov DIG3, r7
    mov DIG2, r7
    mov DIG1, r7
    mov DIG0, r7
    mov COL0, r7
    mov COL1, r7
    mov COL2, r7
    mov COL3, r7
    mov COL4, r7
    mov LD, r7
rts

write:                     ;SCORE
    mov r7, #0x26          ;S betű a pontmátrixra
    mov COL4, r7
    mov r7, #0x49
    mov COL3, r7
    mov COL2, r7
    mov COL1, r7
    mov r7, #0x32
    mov COL0, r7
    mov r7, #0x39
    mov DIG3, r7           ;C
    mov r7, #0x3F
    mov DIG2, r7           ;O
    mov r7, #0x77
    mov DIG1, r7           ;R
    mov r7, #0x79
    mov DIG0, r7           ;E
rts

IE_sub:
    mov r6, #0x08
    mov BTIE, r6           ;A BT3 gomb megszakító tevékenységének engedélyezése
rts

```

```

pause:
    mov r7, #0x7F          ;P betű a pontmátrixra
    mov COL4, r7
    mov r7, #0x09
    mov COL3, r7
    mov COL2, r7
    mov COL1, r7
    mov r7, #0x06
    mov COL0, r7
    mov r7, #0x77
    mov DIG3, r7          ;A
    mov r7, #0x3E
    mov DIG2, r7          ;U
    mov r7, #0x6D
    mov DIG1, r7          ;S
    mov r7, #0xF9
    mov DIG0, r7          ;E

    pause_loop:
    mov r3, BT             ;Buttont bemozgatjuk az r3-as regiszterbe későbbi vizsgálódás céljából
    tst r3, #0x02          ;le van nyomva a BT1 gomb?
    jz pause_loop         ;ha nincs, akkor marad a loop-ban
    mov r4, #0xFF          ;a button interrupt flag feltöltése,
    mov BTIF, r4           ;így működött
    jsr clrscr
    jsr display
    rti                   ;ha meg van nyomva, visszatérés a megszakításból

```

A játék működés közben (1. verzió, még nincs megszakítás és nehézség beállítás, valamint a pontot is lehet veszíteni)

A dokumentációban ismertetett verzió