Mikroelektronika labor

PS2 perifériavezérlő rendszerterv

Készítette: VÉGH GERZSON, Kovács Patrik

2023

## Rendszerterv

A hálózat célja egy periféria vezérlő, mely az ACU softcore processzor és egy PS/2 protokol billentyűzet között teremt kapcsolatot. Az érkező billentyűket megfelelő kis- vagy nagy ASCII karaktérré alakítja, input buffer-ben tárolja és a vezérlés természetéből adódó flagekkel egészíti ki.

#### Flagek:

A flagek mind 0 állapota egy „közönséges” billentyű lenyomását, majd felengedését takarja, amely normál gépelési szokásoknak megfelel.

#### F0:

Amennyiben egy billentyűzetet lenyomtunk, de nem engedtünk el adott időn belül, ez a flag jelzi a folyamatosan lenyomott billentyűt. Ez lehet tudatos felhasználói döntés, vagy meghibásodás miatt beragadt billentyű.

#### F1:

Ha egy olyan billentyűt engedünk el, amelyet nem nyomtunk meg, akkor a flag 1-be áll. Ennek az oka lehet meghibásodás vagy elégtelen roll-over.

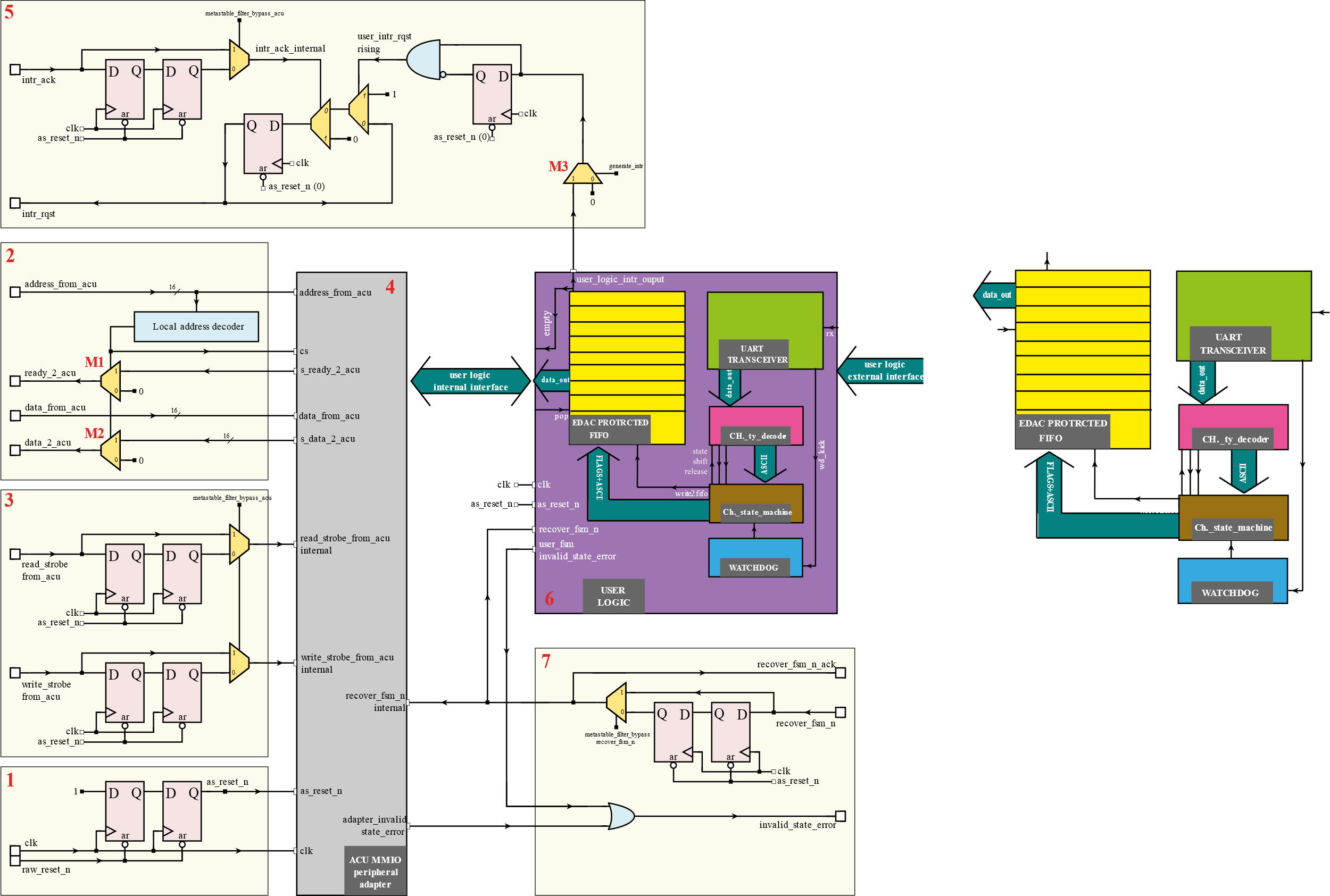
#### F2:

A flag billentyűk elengedése nélküli újabb billentyű lenyomását jelzi. Ez lehet tudatos felhasználó döntés, de nem normál gépelési szokásoknak megfelelő. (A funkcionális gombokra nem vonatkozik).

#### F3-F7 flag-ek nem implementáltak.

### Generikus paraméterek:

Az implementációhoz szükséges két paraméter a saját címek, pontosabban a karakter kiolvasás és az empty flag kiolvasás MMIO címei.



ábra 1 A periféria vezérlő blokkvázlata.

### USER LOGIC Bemenetei:

#### RX:

A PS/2 billentyűzet adatvonala

#### Char\_req:

A peripheral adapter állapotgép által címből lefordított pop utasítás, az elsőre érkezett karakter és flagjeinek kiolvasásához.

### USER LOGIC Kimenetei:

#### Get\_char:

16 bites adatbusz egy ASCII karaktér és 8 bitflag kiolvasására

#### Is\_empty:

A tároló ürességét jelző bit. Ennek negatív átmenete generál megszakítást, valamint címzéssel ki is olvasható.

## Implementáció

### Szintézis paraméterek:

* Érkezett karakter kiolvasás read címe
* Kiolvasható karakter(ek) elérhetősége read cím
* watchdog timeout ideje, az el nem engedett gombok kezelésére

### Az áramkör

#### bemenet:

#### *Rx* az uart bemenete, ps/2 billentyűzet adatvonal kimenete

2. ábra példa hullámforma, ami megjelenhet az rx bemeneten.

A billentyűzet 8o1 protokollal kommunikál.

#### kimenet:

a külvilág felé nincs kimenete

#### Periféria illesztő felépítése

##### UART\_TRANSCEIVER

Az Rx fogadására a szolgáltatott uart\_transceiver hardware modelt használtuk fel, csak fogadó, paritás ellenőrzést letiltva 9 adatbit konfigurációval. A páros paritést a 8+1 ”adat” bitből számoljuk ki. A fogadott adat megjelenik a 8 bites párhuzamos kimenetén.

#### CHARACTER\_TY\_DECODER

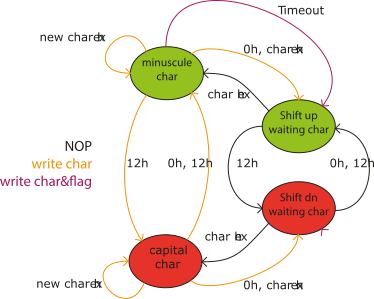
A ps/2 billenytű kód byte-ot a character\_ty\_decoder dolgozza fel, pontosabban lefordítja a gombnak megfelelő kis- vagy nagybetű ascii kódjára, illetve érzékeli a különleges billenytű parancsokat. Ezeket 8 bites buszon és minden egyes spec karakternek saját vezetékén jelzi. Egy további logikai bit bemenete segítségével dől el a betű mivolta. A screen shot of a computer

Description automatically generated with low confidence

ábra 3 a dekóder időzítési diagramja

##### CHARACTER\_STATE\_MACHINE

Ez a core követi nyomon az érkező beviteleket, előéletét állapotváltozóban tárolja majd ezek és az elköveetkező bemenő ascii segítségével dönti el, hogy a 16 bites kimenetén milyen karakter jelenjen meg milyen mellékes flagekkel, valamint, hogy mikor kerüljön karakter beírásra az ideiglenes FIFO tárolóba.



ábra 4 az állapotgép irányító diagramja

##### WATCHDOG

A watchdog figyeli, hogy van-e kommunikáció. Ha szintézisparaméterként megadott idejig nem érkezik új üzenet, a ”beragadt” karakter a megfelelő flag kíséretében kiírásra kerül a CSM által; erre a parancsot egy timeout bit állítása adja ki, és erre egy nyugtázást vár. 

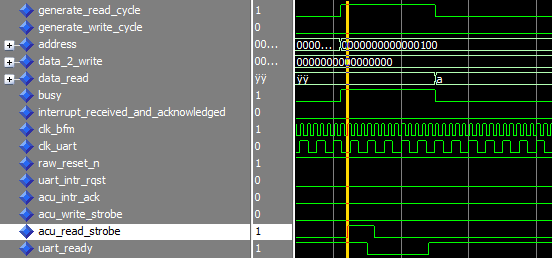
5. ábra A watchdog időzités diagramja.

##### EDAC\_PROTECTED\_FIFO

Az ideiglenes tároló megvalósításásra a szolgáétatott edac\_protected\_fifo IP core-t használtuk fel. Az architektúra szolgáltatásaiból a beírást (CSM által), pop-olást használjuk adat kezelésre. A core empty flagjét felhasználjuk az eltárolt karakterek meglétének jelzésére, ennek negatív átmenetével interrupt jelet generálunk a processzor felé, illetve a flag értéke ki is olvasható, hogy megyőződjünk róla, hogy kiürítettük azt.

##### ACU\_MMIO\_PERIPHERIAL\_ADAPTER FINITE STATE MACHINE

A processzor által kiadott memória műveletek (MMIO) lefordítása a periféria műveleteire a séma állapotgépének a feladata. A két elfogadott művelet reard\_req és flag\_req. Adott címről olvasás műveletre időzítést betartva a FIFO-ból egy 16bites adat (ascii + flagek) kiolvasása történik meg. Másik adott memória címről olvasásra, időzítéseket betartva a FIFO ”ürességéről” kérhetünk információt, logikai érték 16 bitre kiterjesztve. További címekről olvasási és bármilyen írás műveletre nem reagálunk.



6. ábra Olvasási ciklus időzítés diagramja.

***A képen Multimédiás szoftver, szoftver, Grafikai szoftver, szöveg látható

Automatikusan generált leírás***

**7**. ábra A felhasznált erőforrás mértéke (sötét lila)