

blic nije težak, ne mogu se konkretno sjetiti što je točno trebalo, uglavnom moras znati postaviti bitove za ulaz i izlaz na gpio, moras znati postaviti konstantu za usporedbu u rtc ($rtc = 10\text{kHz}$ tebi treba da odbroji 100ms), trebas na jednom mjestu postaviti brojac, na drugom naredbu da ga smanjuješ kako se obavi to što treba, da spojiš dva podatka (8-bitna) u jedan, znači jedan shiftas za 8 pa ovrha sa drugim, napisati naredbu ispitivanje kraja bloka.... to je ono čega se mogu sjetiti

LIVE: Pojasniti LCDWR potprogram (upisuje jedan znak skroz desno u interni registar LCDa a ovaj skroz lijevi znak istisne, kad posaljemo znak #0A ispisu se svi znakovi iz internog registra na display.)

Kako znamo da je LCD spojen na izlaz B (zato jer je po defaultu postavljeno registri smjera za A i za B u nulu a u toj situaciji je A ULAZ kad je u nuli a B je IZLAZ kad je u nuli. Pise na prezentacijama.)

Kako bi napravili da LCD bude spojen na A? (Staviti bitove registra smjera podataka za vrata A u jedan, time ti bitovi postaju izlazni.)

BLIC: Nadopuniti program vezan za GPIO. Po meni njatezi blic do sad...

1. Pitanje: Prvi program DIV potprogram objasniti, i objasniti zašto one uvjete postavljati kad je x negativan da provjerava $<$ a kad je pozitivan $>$

Odgovor: $f(-x) = (-x)^3 - 1 / (2 * (-x)) = (x^3 + 1) / 2x > 0$ za svaki x manji od nula, stoga se ista funkcija može izvoditi samo drugaciji uvjet jer ide iz negativnog prema nuli,

A kad je $x = 0$, funkcija je trivijalno rješena i rješenje je postavljeno u nulu (bar kod mene)

uživo s labosa:

objasniti kako funkcionira naredba: LDRB R2, [R4,R3]

Od asistenata su bili knezovic i neki random tip. Mene je ispitivo taj drugi, trebao sam objasniti kak se dijeli u prvom, kak funkcioniraju potprogrami, i sta znaci kad ubacimo "A" (line feed ?) u LCD. Mislim da niko nije frknuo van pa su valjda oboje dobri 🍷

A blic...

- trebalo je napisati i ubaciti konstantu u registar za usporedbe RTC-a (prekid se mora događati svakih 100 ms sa frekv. 10kHz tak da konstanta onda bude 1000 dekadski)
- staviti 0 u registar za brojanje, da pocne brojati od 0 (RTC)
- staviti 1 u onaj registar da se enablaju prekidi (RTC)
- enablat prekide u CPSR-u, trebalo je napisati ono BIC r0 r0 #80
- u prekidnom potprogramu učitati GPIO adresu
- spojiti visih 8 bitova (sa A vrata) i nizih 8 bitova (sa B vrata) u jedan reg. pomicanjem, zbrajanjem

Izlazni je bio sa RTC i GPIO, cak niti lcd nije bio u tom slucaju, ali opet slicno kao 2. zadatak iz samog labosa... trebalo je učitati 16B NBC broj na nacin da je visih 8B iz vrata A, a nizih 8B iz vrata B, a prekid se radio svakih 100ms na frekvenciji 10kHz. i taj 16B NBC broj je trebalo onda spremati na neku adresu.

ugl, u kodu je trebalo nadopuniti one tehnikalice tipa:

MRS R0, CPSR

BIC R0, R0, #80 <- ovo je trebalo, a moglo se viditi iz koda da je ORG 18 pa da je IRQ pa je zato #80

MSR CPSR_c, R0

bilo je i za nadopuniti one stvari koje se gledaju po salabahteru (tipa da RTC radi prekid, pa stavis 1 u onu njegovu zadnju adresu)

na kraju programa su bile RTC i GPIO labele + labela od konstante, a konstanta je bio broj koji je trebalo upisati u registar usporedbe od RTC (to se trebalo racunati kao $0.1 * 10 * 10^3$, pa je to 1000, naravno trebalo je %D 1000).

kod vecine linija su bili komentari pa se znalo sta se treba napisati, a ono spajanje u 16B NBC je trebalo napraviti da kad se loada 8 bitova iz A, da se to shifta ulijevo za 8, i onda se taj registar ORRa sa drugim registrom u kojem je loadano nizih 8 bitova iz B - tako onda dobite 16 bitni broj di su gornjih 8 bitova iz A, a donjih iz B.

sto se tice blica, zadatak bio 8 bitni 2k broj se dobije s A mislim, treba provjerit jel pozitivan i ako je na lcd ispisat + i dodat u zbroj pozitivnih na nekoj adresi, ako je negativan ispisat - na lcd i povecat broj negativnih primljenih podataka, rtc radi s nekom frekvencijom vamo tmao, postavlja prekid.. ne sjecam se svega točno, od popunjavanja je bilo:

u prekidnom potprogramu:

spremit kontekst, učitat broj, onda jos nesto ceg se ne sjecam, obnovit kontekst, vratit se iz prekidnog potprograma

u glavnom programu:

postaviti konstantu za rtc, napraviti beskonacnu petlju, i ima ono LCDWR potprogram u kojem na kraju fali vraćanje iz potprograma

to je ubiti svih 8 crtica koje su se trebale nadopuniti, bilo je viska crta pa ti napises di sta mislis koliko treba i zadatak je po mom misljenju bio dosta slican ovom drugom iz vježbe

Što se blica tiče po meni je bilo malo vremena. Ovo su najčešće pogreške bile koliko sam primjetio:

-U prekidnom potprogramu trebalo je sačuvati sadržaj registra(staviti registre na stog) naredbom : STMFD R13,{"Registri koji se koriste u prekidnom potp."},R14} dok ljudi iz nekog razloga su naveli samo jedan registar i R14 ili nisu naveli uopce ni R14(njega je potrebno spremiti jer se koriste dodatni potprogrami u tom prekidnom programu).

- Druga najčešća greška je bila to što mnogo ljudi(a tako i ja) nismo se sjetili(ili znali uopce) naredbu BIC. Sa naredbom BIC trebalo je omogućiti prekide.

-Treća DEFINITIVNO najčešća greška je bila u BROJACU. Procesor je radio na 10 kHz i svakih 50 ms je trebao postavljati prekid. Više manji svi smo zaboravili na tih 50 ms pa smo pisali DW %D 10000.

Sve u svemu, zeznuli su nas sa vremenom pošteno, 10 min svakako nije dovoljno.

Bio je GPIO.

Trebalo je učitati adresu u registar, konfigurirati vrata B tako da jedan bit bude READY, drugi STROBE, pa nakon što se na strobe pošalje 1, trebalo je nadopuniti da se poslije šalje 0, učitavati brojeve sa neke adrese s time da se onda adresa mora povećati za 4, napraviti uvjet za petlju i slične stvarčice.

Iako većina ovdje govore da vremena nije dovoljno, ja smatram da jest, samo treba znati to što radite.

A sto se tice testa.

Bio je neki program koji s porta A čita 6-bitni podatak, predznacno ga prosiruje na 32-bitna i ako je negativan povećava brojac i sprema na lokaciju BROJ. Uz svako ispitivanje ispisuje na LCD ako je negativan pise "+", ako je negativan "-".

6 i 7 bit su sinkronizacijski bili pa je trebalo ispitivati spremnost portova. (Znaci tipa ANDS, R0, R0, #40) .Ugl skoro kraj svake crte je bio komentar gdje je pisalo sto treba raditi.

Bilo je 8 crta i jedna mi je ostala prazna. Samo proucite 2. zadatak dobro i sve ok ;)

u izlaznom je bio zadatak s GPIO i RTC. GPIO je primao 8 bitni podatak i u prekidnom je trebao prebrojiti jedinice u njemu. Nakon što je prebrojao jedinice, trebalo je poslati broj jedinica na LCD. Tako da se pretvori u ASCII i pošalje na LCD koji je spojen na PIO.

Mi smo trebali dopuniti:

1. Spremiti kontekst (sve registre, jako ih je puno)
2. Brojanje jedinica (ADD s uvjetom)
3. Pretvoriti broj jedinica u ASCII
4. Provjeriti dal je broj paran (shifatati i zastavica CC)

U glavnom:

1. Ucitati adresu RTC
2. Omogućiti prekide (BIC r0, nešto)
3. još nešto ????
4. vratiti signal zadnjeg bita u nulu u podprogramu LCDWR
5. izračunati koju konstantu treba staviti u registar usporedbe RTC-a

Evo vam TOČNA rješenja, potvrđeno sa njihovih papira izlaznog blica... prvih 6 linija kôda ostale nisam stigao... Radi se o onom sa GPIO varta A i B podatkom kad je paran/neparan

Source kod:

```
LDR R2, GPIO
STRB R1, [R2, #0C]
LDRB R1, [R2]
AND R0, R0, #0F
ANDS R0, R1
BEQ L2
```

3 boda od mene, peace out za ARH1 labose ove godine :)

blic: ispisivati na ekran + ili - ovisno je li broj poz il neg, spremati negativne na lokaciju BROJAC, i ti brojevi su 6-bitni pa ih se treba predznačno proširiti (to trebaš napisati na crtju). treba spremati brojač na lokaciju BROJAC, treba provjeriti spremnost, treba definirati adresu GPIO-a s DW 0FFFF0000 (koja već piše u tekstu zadatka), treba inicijalizirati GPIO (kao u onim zadacima u slideovima). pogledajte i one kodove na slideovima i znat ćete

Jutros blic-citaju se 2-k brojevi potrebno je učitati bajt sa LDRSB, zatim Spremanje i vraćanje konteksta,obavezno spremati R14, napraviti besk.petlju, učitati konst za RTC i učitati podatak sa GPIO, adresa zadana i to je otprilike to

U blicu je trebalo učitati broj sa porta a i ispitivat jedinice, zatin na portu b su signali ready i strobe, zapamtite da je ready ulazni pa trebate staviti jednicu na b portu(upisati#2) , u djelu zadatka gdje broji jedinice i oduzima od brojaca koji je 8 uvjet je BNE L1.

E sad, trebalo je ispitat najnizi bit, nakon cega je u programu taj podatak bio shiftan u desno za jedan, ja sam to napravio tako da comperam sa 0 i priznala mi je, iako mislim da ne bi to radilo, a ono su to napravili tako su koristili naredbu SUBS pa cudo nekin registara shifta i ostalog 🤖

Na pocetku programa trebalo je samo loudat brojac podataka u jedan registar, a prije toga storat 1 na strobe, i nakon toga storat 0 na ready i strobe..

Eto toliko sam zampamtio, nadam se da ce nekome pomoc 🤖

a izlazni:

moras inicijalizirat PIO(vrata A i vrata B) i vrsit upisivanje rezultata kruzno.... dosta ih je il potjerano il su otisli s nulom....od nekih 12 ljudi ostalo pisat 5 :/ srednja zalost =)