Građa računala

Vježbe 11
Iznimke i prekidi

Materijali

- Knjiga
  - Pregled – poglavlje 14
  - Detalji – poglavlje 15
- Easy68K
  - Help -> Exceptions
- Web
  - http://research.cs.tamu.edu/prism/lectures/mbsd/mbsd_l9.pdf

User vs. Supervisor

- Većina programa radi u korisničkom načinu
  - Veća sigurnost – korisnik nema sva prava
- Povremeno korisnik treba koristiti "opasne" instrukcije
  - Pisanje po sistemskoj memoriji – nužno za ispis na ekran ili u datoteku
- Potrebno mu je omogućiti prijelaz u nadgledni način rada
  - Korisniku je zabranjeno direktno mijenjanje statusnog registra!

- Prijelaz Supervisor -> User
  - Mijenjanjem SR – MOVE, AND, OR

- Prijelaz User -> Supervisor
  - Iznimke/prekidi

Iznimke i prekidi

- Iznimka – nešto što mijenja normalni redosljed izvršavanja instrukcija
  - Poziv metoda OS-a *(OS call)* – izvršavanje "opasnih" operacija u sigurnom kontekstu
  - Ideja: korisnik je ograničen na ponuđene metode
- Slično branchu/pozivu funkcija ali:
  - Ne mora biti eksplicitnog poziva
  - Mogu se potencijalno dogoditi bilo gdje i bilo kada
  - Adresu nije potrebno prosljediti – adrese su *preddefinirane* od strane sistemskog programera (tj. OS-a)
  - Sprema se više podataka na sistemski stog
    - uz PC barem još SR
- Prekid – iznimka uzrokovana izvan procesora (hardverom)
  - Unos pomoću tipkovnice
  - Iscrtavanje na ekran itd.

Obrada iznimki

- Pomoću *exception handlera*
  - Svaki tip iznimke ima pridruženu adresu na kojoj se nalazi adresa koda za obradu iznimke - *handler*
  - Lokacije adresa handlera su preddefinirane
    - *tablica vektora*
  - Tablicu je potrebno definirati u nadglednom načinu, prije pokretanja korisničkih programa (općenito to radi OS)
  - RTE instrukcija – za povrat iz nekih iznimki
    - Očekuje PC i SR na vrhu sistemskog stoga
- Samo izvršavanje se vrši u nadglednom načinu rada procesora
  - pristup svim instrukcijama
  - nakon poziva, korisnik nema kontrolu nad izvršavanjem

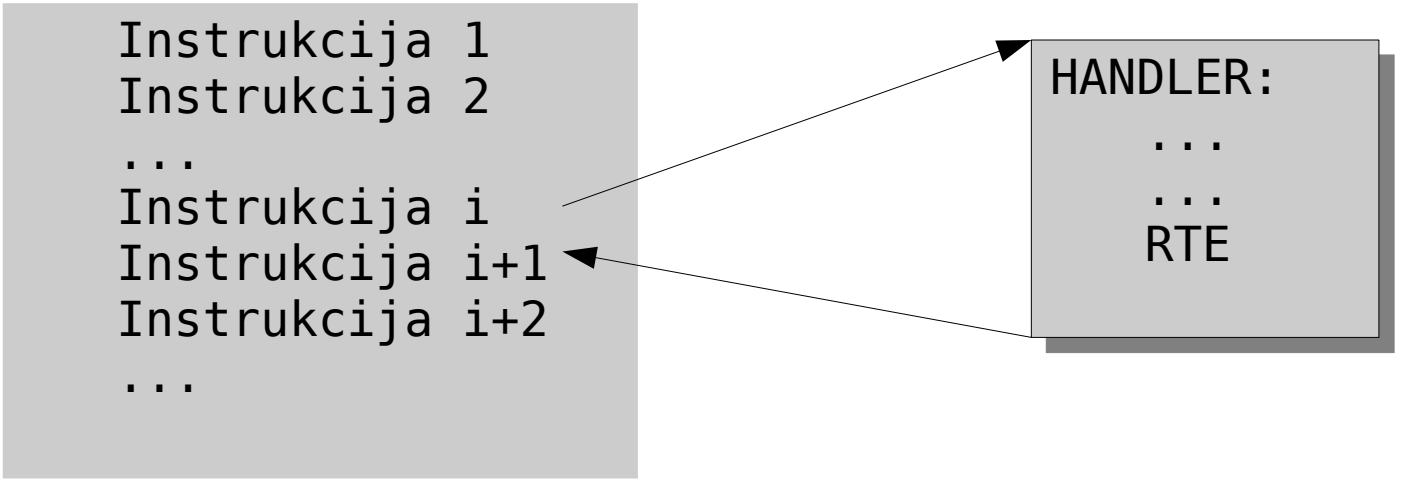
Tablica vektora

Vector Number	Offset	Assignment
0	000	Reset: Initial interrupt stack pointer
1	004	Reset: Initial program counter
2	008	Bus error
3	00C	Address error
4	010	Illegal instruction
5	014	Divide by zero
6	018	CHK, CHK2 instruction
7	01C	cpTRAPcc, TRAPcc, TRAPV instruction
8	020	Privilege violation
9	024	Trace
10	028	A-line emulator
11	02C	F-line emulator
12	030	Reserved
13	034	Coprocessor protocol violation
14	038	Format error
15	03C	Unitialized interrupt
16-23	040-05C	Reserved
24	060	Spurious interrupt
25	064	Autovector (level 1)
26	068	Autovector (level 2)
27	06C	Autovector (level 3)
28	070	Autovector (level 4)
29	074	Autovector (level 5)
30	078	Autovector (level 6)
31	07C	Autovector (level 7)
32-47	080-0BC	TRAP #0-15
48	0C0	FPCP Branch or set on unordered condition
49	0C4	FPCP Inexact result
50	0C8	FPCP Divide by zero
51	0CC	FPCP Underflow
52	0D0	FPCP Operand error
53	0D4	FPCP Overflow
54	0D8	FPCP Signaling NAN
55	0DC	Reserved
56	0E0	PMMU configuration
57	0E4	PMMU illegal operation
58	0E8	PMMU access level
59-63	0EC-0FC	Reserved
64-255	100-3FC	User defined vectors

FPCP=floating point coprocessor
PMMU=paged memory management unit

- Adrese $0 - $3FF
- Svaki "redak" tablice sadrži 4 bajta
- Ta 4 bajta tvore adresu prve instrukcije handlera za pojedinu iznimku
- Svaka iznimka ima pridružen jedinstveni indeks u ovoj tablici

Obrada iznimki

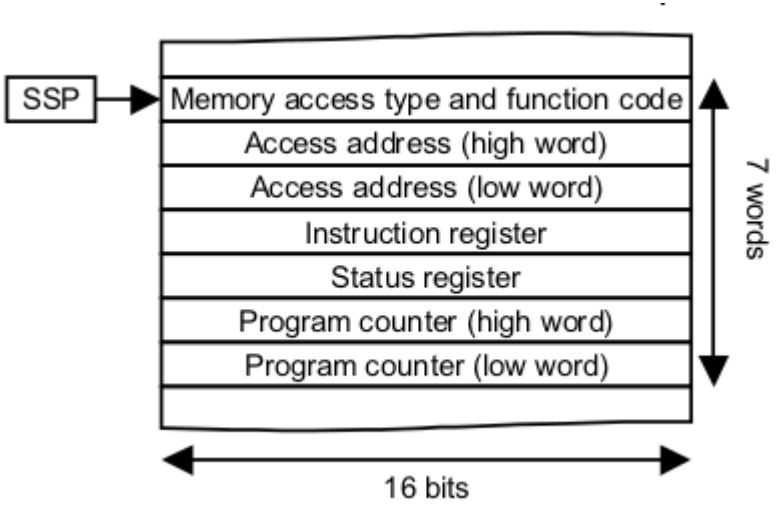


- Kada instrukcija ***i*** izazove iznimku
  - Prvo se dovršava izvršavanje instrukcije (izuzeci ovom pravilu kasnije)
  - Pohranjuje se PC (ovisno o iznimci, može se pohraniti još podataka) na sistemski stog
  - Pohranjuje se SR na sistemski stog
  - Dohvaća se indeks vektora iz kojeg se izvodi adresa *handlera*: adresa = 4 * index
  - Supervisor bit postavlja se na 1, trace bit na 0

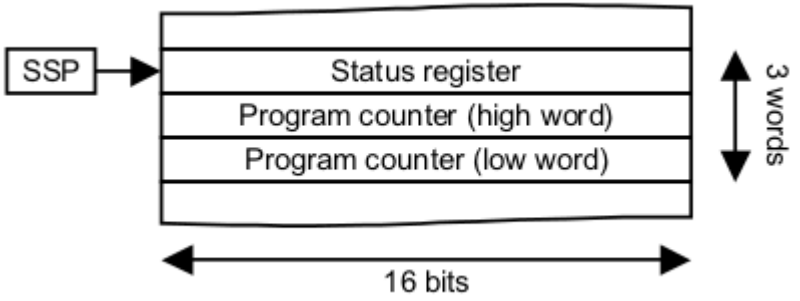
- Kada je iznimka obrađena
  - Sa stoga se dohvaćaju svi pohranjeni parametri – RTE nije uvijek dovoljan!
  - Izvršavanje se vraća na instrukciju ***i***+1
  - VAŽNO: program bi trebao nastaviti kao da se iznimka nije dogodila (osim kad iznimka **mora** utjecati na izvršavanje)

Tipovi iznimki

Kategorija	Iznimke	Broj **riječi** na stogu
0	Reset Bus error Address error	7
1	Trace Interrupt Illegal op-code Privilege	3
2	TRAP TRAPV CHK DIV-BY-ZERO	3



Kategorija 0



Kategorije 1 i 2

Interrupt

- Kada neka komponenta računala želi komunicirati s procesorom, šalje prekid (interrupt)
- Prekidi su organizirani hijerarhijski
  - 7 razina
  - *Interrupt bit mask* – u statusnom registru
  - Ukoliko je razina prekida manja ili jednaka interrupt bit mask, prekid se ignorira
- Ostalo na sljedećim vježbama!

TRAP instrukcija

- Instrukcija za uzrokovanje iznimki
  - Omogućuje pozivanje sistemskih metoda bez da "kršimo pravila"
  - "neiznimne" iznimke jer spadaju u standardni rad procesora
- Vektori 32 – 47 (dekadski)
  - 16 vektora ali uz mogućnost prenošenja dodatnih parametara u registrima
  - TRAP #0 – TRAP #15
- TRAP #15 – preddefinirane iznimke za lakši rad s emulatorom (specifično za Easy68k emulator)
  - Text I/O
  - File I/O
  - Network I/O itd.

TRAP - primjer

- Ispis teksta u konzolu
  - TRAP #15
  - U D0 stavljamo #13
  - Ispisuje C-ovski string na lokaciji zadanoj u A1

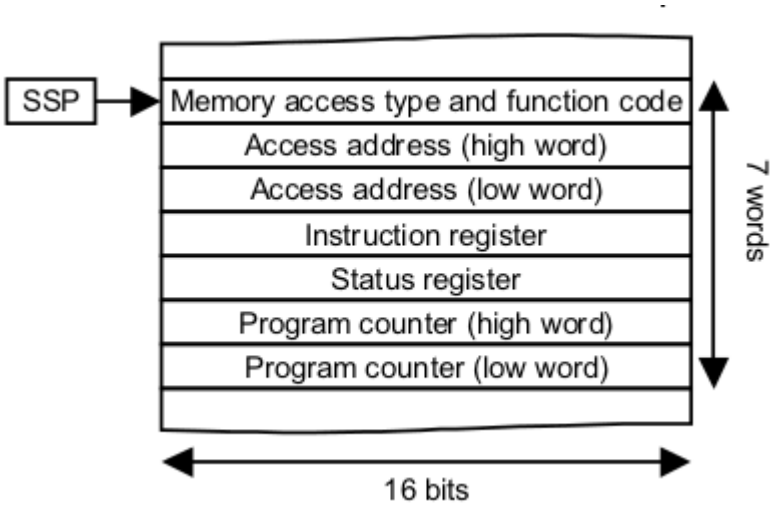
```
       ORG $1000
ISPIS: DC.B  'String mora imati nulu na kraju!', 0

       ORG $1200
START:
       MOVE.L #13, D0
       LEA    ISPIS, A1
       TRAP   #15
       SIMHALT
       END    START
```

ADDRESS

- Kada želimo spremati/čitati word ili long na neparnoj adresi

```
        ORG $1200
START:
        MOVE.L #123, $9005
        END    START
```



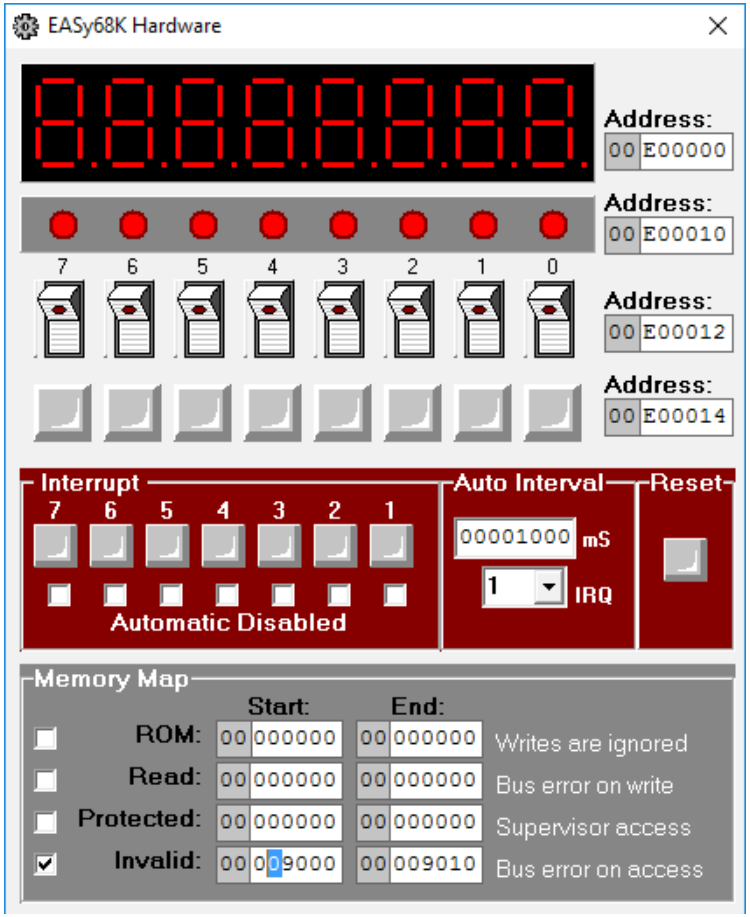
Primjer sistemskog stoga:

```
00FFFFC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFF0: FF FF 00 15 00 00 90 05 23 FC 20 00 00 00 12 0A --------#- -----
```

BUS

- Kada sabirnica ne može završiti ciklus, odnosno pribaviti/poslati podatke
- U Easy68k potrebno je prvo podesiti "hardver" da bismo mogli generirati ovu iznimku
- Nakon toga:

```
        ORG $1200
START:
        MOVE.L #123, $9004
        END    START
```



Divide-by-zero

- Kada se dijeli s nulom

  Instrukcije:
  - DIVS, DIVU

```
; generiranje iznimke
    ORG $1000
START:
    DIVS  #0, D0
    END   START
```

```
00FFFFC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------
00FFFFF0: FF FF FF FF FF FF FF FF FF FF 20 00 00 00 00 04 ---------- -----
```

ILLEGAL

- Kada se pokuša prevesti kod koji ne odgovara niti jednog instrukciji procesora

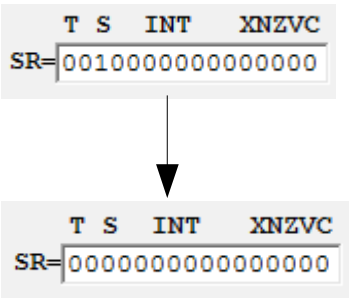
```
        ORG $1200
START:
ILLL:   MOVE.W    #$EFFF, ILLL
        JMP       ILLL
        END       START
```

- Pažnja!
  - Gornji kod korumpira memoriju
  - reset emulatora neće pomoći (potrebno je ugasiti i ponovno upaliti emulatorski prozor)

PRIVILEGE

- Kada korisnik pokušava koristiti instrukcije koje pripadaju nadglednom načinu

```
; generiranje iznimke
    ORG $1000
START:
    MOVE   #0, SR
    MOVE   #0, SR
    END    START
```

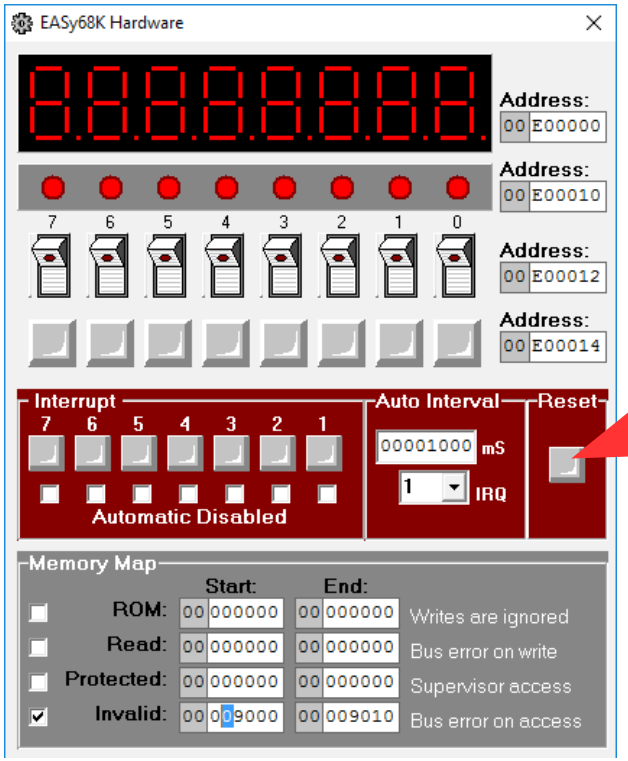


- Prva instrukcija poništava nadgledni način pa druga ne može modificirati statusni registar
- Povlaštene instrukcije:
  - Modificiranje SR-a (and, or, eor, move)
  - Modificiranje SSP-a (move)
  - RESET, RTE, STOP

RESET

- Kod paljenja/gašenja
- Kada se dogodi greška koju sustav ne može popraviti
- Događa se sljedeće:
  - S = 1, T = 0

    onemogućuju se daljnji prekidi
  - Vrijednost adrese $0 -> SR
  - Vrijednost adrese $4 -> PC



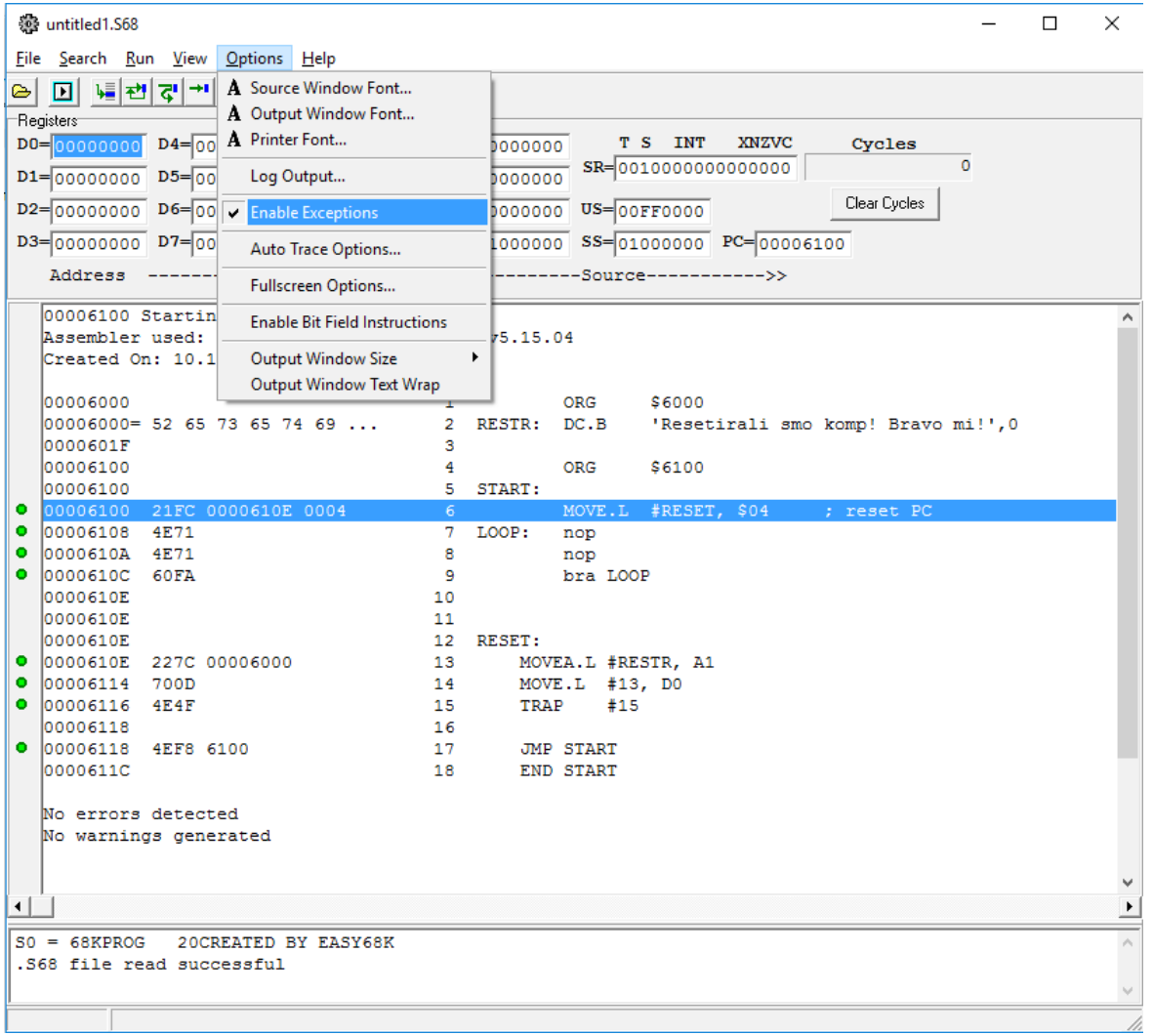
TRACE

- TRACE bit u statusnom registru
  - Ako je 1, obrađuje se iznimka na vektoru $24
  - Iznimka se izvršava nakon svake instrukcije!

- Korištenje:
  - Za *debagiranje*

Primjeri rada s iznimkama

- Prvo, uključiti "**enable exceptions**" u prozoru **simulatora** prije pokretanja programa
  - U suprotnom program staje s izvršavanjem kad naiđe na iznimku



Definiranje vlastitih metoda obrade

1) Options -> Enable Exceptions
2) Pogledati indeks iznimke koju želimo obrađivati
3) Na adresu indeks * 4 stavimo adresu prve instrukcije koda koji obrađuje iznimku
4) Napisati kod za obradu
5) Napisati kod za povratak na normalan način izvršavanja – posebno paziti da na sistemskom stogu ne ostanu zaostali podaci!