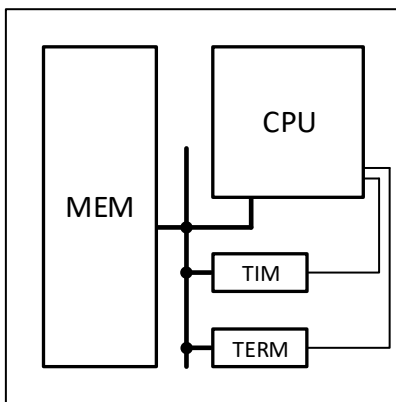


OPIS RAČUNARSKOG SISTEMA

Računarski sistem se sastoji od procesora, operativne memorije, tajmera i terminala. Sve komponente računarskog sistema su međusobno povezane preko sistemske magistrale. Tajmer i terminal, kao periferije, su povezani sa procesorom i preko linija za slanje zahteva za prekid. Slika 1. predstavlja uprošćen šematski prikaz posmatranog računarskog sistema.



Slika 1. Šematski prikaz računarskog sistema

Opis procesora

U nastavku je opisan deo 16-bitnog dvoadresnog procesora sa Von-Neuman arhitekturom. Adresibilna jedinica je jedan bajt, a raspored bajtova u reči je little-endian. Veličina memorijskog adresnog prostora je 2^{16} B. Počev od adrese $0 \times \text{FF}00$ memorijskog adresnog prostora nalazi se prostor veličine 256 bajtova rezervisan za memorijski mapirane registre (registri kojima se pristupa instrukcijama za pristup memorijskom adresnom prostoru). Počev od adrese 0×0000 memorijskog adresnog prostora nalazi se IVT (interrupt vector table) sa osam ulaza. Svaki ulaz zauzima dva bajta i sadrži adresu odgovarajuće prekidne rutine. Ulazi u IVT odgovaraju sledećim prekidnim rutinama:

- ulaz 0 sadrži adresu prekidne rutine koja se izvršava prilikom pokretanja odnosno resetovanja čitavog procesora (ne izvodi se kompletna sekvenca obrade prekida već se samo vrši skok na adresu koja se nalazi u okviru datog ulaza),
- ulaz 1 sadrži adresu prekidne rutine koja se izvršava ukoliko se pokuša izvršavanje nekorektne instrukcije (nepostojeći operacioni kod, neispravan način adresiranja itd.),
- ulaz 2 sadrži adresu prekidne rutine koja se izvršava kada stigne zahtev za prekid od tajmera (opis principa rada tajmera i način njegove konfiguracije dat je u zasebnom poglavlju),
- ulaz 3 sadrži adresu prekidne rutine koja se izvršava kada stigne zahtev za prekid od terminala (opis principa rada terminala dat je u zasebnom poglavlju) i
- ostali ulazi su slobodni za korišćenje od strane programera.

Procesor poseduje osam opštenamenskih 16-bitnih registara označenih sa $r\langle \text{num} \rangle$ gde $\langle \text{num} \rangle$ može imati vrednosti od nula do sedam. Moguće je zasebno koristiti viših ili nižih osam bita svakog od opštenamenskih registara kao 8-bitni registar označen sa $r\langle \text{num} \rangle h$ ili $r\langle \text{num} \rangle l$, respektivno. Registar $r7$

se koristi kao `pc` registar (pokazuje na instrukciju koja se u memoriji nalazi neposredno iza trenutno izvršavane instrukcije). Registar `r6` se koristi kao `sp` registar (pokazuje na zauzetu lokaciju na vrhu steka, a stek raste ka nižim adresama). Pored opštenamenskih registara postoji `psw` registar (statusna reč procesora).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	Tl	Tr										N	C	O	Z

Značenje flegova u `psw` registru:

- **Z (Zero)** - rezultat prethodne operacije je nula,
- **O (Overflow)** – prekoračenje,
- **C (Carry)** - prenos,
- **N (Negative)** - rezultat je negativan,
- **Tr (Timer)** - maskiranje prekida od tajmera (0 - omogućen, 1 - maskiran),
- **Tl (Terminal)** - maskiranje prekida od terminala (0 - omogućen, 1 - maskiran) i
- **I (Interrupt)** - globalno maskiranje spoljašnjih prekida (0 - omogućeni, 1 - maskirani).

Instrukcije mogu biti veličine od jedan do sedam bajtova. Instrukcija u najopštijem slučaju ima sledeći format:

I	II	III	IV	V	VI	VII
InstrDescr	Op1Descr	Im/Di/Ad	Im/Di/Ad	Op2Descr	Im/Di/Ad	Im/Di/Ad

Prvi bajt instrukcije sadrži operacioni kod i dodatne informacije o instrukciji. Naredni bajtovi instrukcije koriste se za kodiranje operandada. Pojedinačni operand može zahtevati jedan, dva ili tri bajta za kodiranje u zavisnosti od načina adresiranja. Detaljan opis `InstrDescr` i `Op<num>Descr` bajtova instrukcije dat je u nastavku.

7	6	5	4	3	2	1	0
OC ₄	OC ₃	OC ₂	OC ₁	OC ₀	S	Un	Un

Značenje bitova `InstrDescr` bajta instrukcije:

- **OC₄OC₃OC₂OC₁OC₀** - operacioni kod instrukcije,
- **S (Size)** - veličina operandada instrukcije (0 - jedan bajt; 1 - dva bajta) i
- **Un (Unused)** - neiskorišćeni bitovi koji imaju fiksnu vrednost nula.

7	6	5	4	3	2	1	0
AM ₂	AM ₁	AM ₀	R ₃	R ₂	R ₁	R ₀	L/H

Značenje bitova `Op<num>Descr` bajta instrukcije:

- **AM₂AM₁AM₀** - kodiran način adresiranja pri čemu adresiranje može biti:
 - **0x0** - neposredno; vrednost operandada je kodirana u okviru instrukcije pomoću jednog ili dva `Im/Di/Ad` bajta u zavisnosti od veličine operandada; neposredno adresiranje nije validan način adresiranja za destinacioni operand,

- 0x1 - registarsko direktno; vrednost operanda nalazi se u registru čiji je broj kodiran u okviru instrukcije (nema Im/Di/Ad bajtova),
- 0x2 - registarsko indirektno bez pomeraja; vrednost operanda nalazi se u memoriji na adresi ukazanoj vrednošću registra čiji je broj kodiran u okviru instrukcije (nema Im/Di/Ad bajtova),
- 0x3 - registarsko indirektno sa 8-bitnim označenim pomerajem; vrednost operanda nalazi se u memoriji na adresi ukazanoj zbirom vrednosti registra, čiji je broj kodiran u okviru instrukcije, i vrednosti koja se nalazi u jednom Im/Di/Ad bajtu,
- 0x4 - registarsko indirektno sa 16-bitnim označenim pomerajem; vrednost operanda nalazi se u memoriji na adresi ukazanoj zbirom vrednosti registra, čiji je broj kodiran u okviru instrukcije, i vrednosti koja se nalazi u dva Im/Di/Ad bajta i
- 0x5 - memorijsko; vrednost operanda nalazi se u memoriji na adresi ukazanoj vrednošću koja se nalazi u dva Im/Di/Ad bajta,
- R₂R₁R₀ - kodiran broj korišćenog registra (psw registar se kodira vrednošću 0xF) i
- L/H (Low/High) - naznaka da li se koristi nižih ili viših osam bita registra (0 - nižih; 1 - viših) u slučaju registarskog direktnog adresiranja za operand veličine jednog bajta.

Mnemonik	OC	Efekat	Flegovi koji se menjaju
halt	1	Zaustavlja izvršavanje instrukcija	-
xchg dst, src	2	temp<=dst; dst<=src; src<=temp;	-
int dst	3	push psw; pc<=mem16[(dst mod 8)*2];	-
mov dst, src	4	dst<=src;	Z N
add dst, src	5	dst<=dst+src;	Z O C N
sub dst, src	6	dst<=dst-src;	Z O C N
mul dst, src	7	dst<=dst*src;	Z N
div dst, src	8	dst<=dst/src;	Z N
cmp dst, src	9	temp<=dst-src;	Z O C N
not dst	10	dst<=~dst;	Z N
and dst, src	11	dst<=dst&src;	Z N
or dst, src	12	dst<=dst src;	Z N
xor dst, src	13	dst<=dst^src;	Z N
test dst, src	14	temp<=dst&src;	Z N

<code>shl dst, src</code>	15	<code>dst<=dst<<src;</code>	Z C N
<code>shr dst, src</code>	16	<code>dst<=dst>>src;</code>	Z C N
<code>push src</code>	17	<code>sp<=sp-2;</code> <code>mem16[sp]<=src;</code>	-
<code>pop dst</code>	18	<code>dst<=mem16[sp];</code> <code>sp<=sp+2;</code>	-
<code>jmp dst</code>	19	<code>pc<=dst;</code>	-
<code>jeq dst</code>	20	<code>if (equal_condition_is_met)</code> <code>pc<=dst;</code> <code>end_if</code>	-
<code>jne dst</code>	21	<code>if (not_equal_condition_is_met)</code> <code>pc<=dst;</code> <code>end_if</code>	-
<code>jgt dst</code>	22	<code>if (signed_greater_condition_is_met)</code> <code>pc<=dst;</code> <code>end_if</code>	-
<code>call dst</code>	23	<code>push pc;</code> <code>pc<=dst;</code>	-
<code>ret</code>	24	<code>pop pc;</code>	-
<code>iret</code>	25	<code>pop psw;</code> <code>pop pc;</code>	psw

Dodatne napomene:

- sve aritmetičke operacije se izvode tako da odgovaraju označenim celim brojevima,
- iza mnemonika asemblerske naredbe, bez belih znakova, može se navesti sufiks `b` ili `w` kako bi se naznačila veličina operanada date instrukcije,
- instrukcije `cmp` i `test` nigde ne čuvaju direktni rezultat odgovarajuće operacije, već samo u skladu sa rezultatom postavljaju nove vrednosti flegova u `psw` registru i
- kombinacije instrukcija i operanada, za koje ne postoji razumno tumačenje, smatrati greškom.

Sintaksa operanada u okviru asemblerskih naredbi:

- `<val>` - neposredno adresiranje vrednosti `<val>`
- `&<symbol_name>` - neposredno adresiranje vrednosti simbola `<symbol_name>`
- `r<num>` - registarsko direktno adresiranje
- `r<num>[<val>]` - registarsko indirektno sa označenim pomerajem
- `r<num>[<symbol_name>]` - registarsko indirektno sa označenim pomerajem
- `$<symbol_name>` - `pc` relativno adresiranje simbola `<symbol_name>`
- `<symbol_name>` - apsolutno adresiranje simbola `<symbol_name>`
- `*<val>` - apsolutno adresiranje podatka u memoriji na adresi ukazanoj vrednošću `<val>`

Primer kodirane instrukcije `xchgw [sp], 0x1234` jeste:

`0x14 0x4C 0x00 0x34 0x12`

Opis terminala

Terminal predstavlja ulazno/izlaznu periferiju koja se sastoji od displeja i tastature. Terminal poseduje dva memorijski mapirana registra. Na adresi `0xFF00` memorijskog adresnog prostora nalazi se `data_out` registar izlaznih podataka. Upisom vrednosti u `data_out` registar na tekućoj poziciji displeja ispisuje se znak koji prema ASCII tabeli odgovara upisanoj vrednosti. Na adresi `0xFF02` memorijskog adresnog prostora nalazi se `data_in` registar ulaznih podataka. Kada se pritisne neki taster (1) upisuje se ASCII kod pritisnutog tastera u `data_in` registar i (2) terminal, kao periferija posmatranog procesora, generiše zahtev za prekid (u okviru prekidne rutine, koja obrađuje ovaj zahtev za prekid, čitanjem vrednosti `data_in` registra može se saznati koji taster je pritisnut).

Opis tajmera

Tajmer kao periferija periodično generiše zahtev za prekid. Perioda generisanja zahteva za prekid definisana je sadržajem `timer_cfg` konfiguracionog registra tajmera. Registar `timer_cfg` je memorijski mapiran registar i nalazi se na adresi `0xFF10` memorijskog adresnog prostora. Njegova inicijalna vrednost nakon pokretanja odnosno resetovanja računarskog sistema jeste `0x0000`.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													T_2	T_1	T_0

Perioda generisanja zahteva za prekid u zavisnosti od $T_2T_1T_0$ vrednosti je sledeća: `0x0` -> 500ms, `0x1` -> 1000ms, `0x2` -> 1500ms, `0x3` -> 2000ms, `0x4` -> 5000ms, `0x5` -> 10s, `0x6` -> 30s i `0x7` -> 60s.