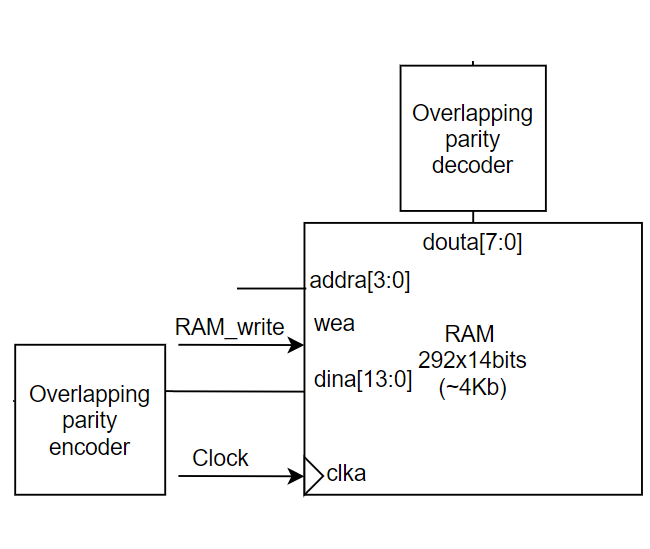
Projekat iz predmeta: Digitalni sistemi otporni na otkaz

Tema projekta:

 Overlapping parity, žaštita memorije u EC1 procesoru

Student: Aleksandar Komazec Mentor: Rastislav Struharik

Dana \_\_\_\_\_\_\_\_\_\_ u Novom Sadu

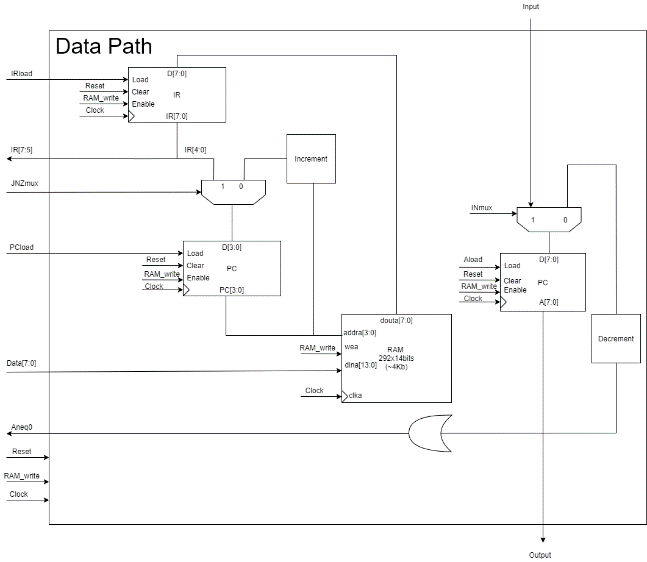
Potpis studenta:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Potpis mentora:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Uvod

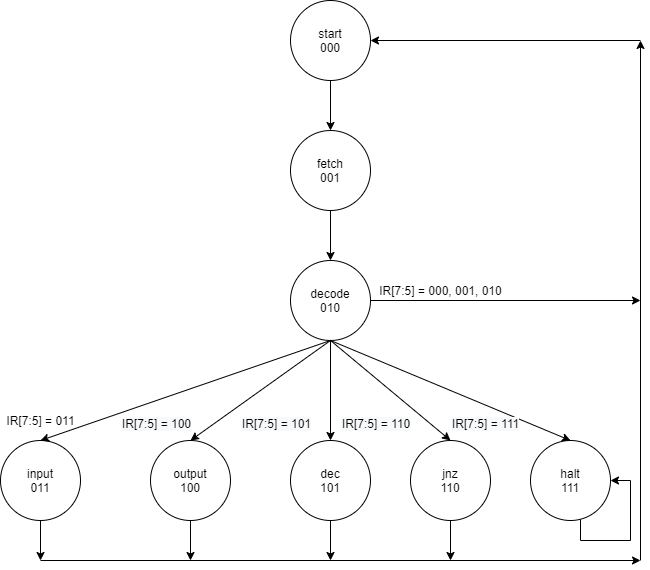
Na Easy computer 1 (ECU 1), jednostavanom osmobitnom procesoru je primenjen overlapping parity. Kako bi se ova tehnika primenila na najjefikasniji način, osmobitna magistrala je podeljena na dva dela po četiri bita I na taj način je predstavljen Hamingov kod (7,4).

1.1 Easy computer one

Easy computer one se sastoji iz dva glavna dela. Prvi deo obezbeđuje tokove podataka (eng. *Datapath Slika 1*).



Slika 1: Data Path

Drugi deo se sastoji od registara i funkcionalnih jedinica i upravljačkog dela (eng. *control unit Slika 2*) koji obezbeđuje upravljanje radom mikroprocesora.

Slika 2: Control Unit

Implementirano je 5 instrukcija: input, output, dec, jnz i halt (Tabela 1)

Tabela 1: Skup instrukcija

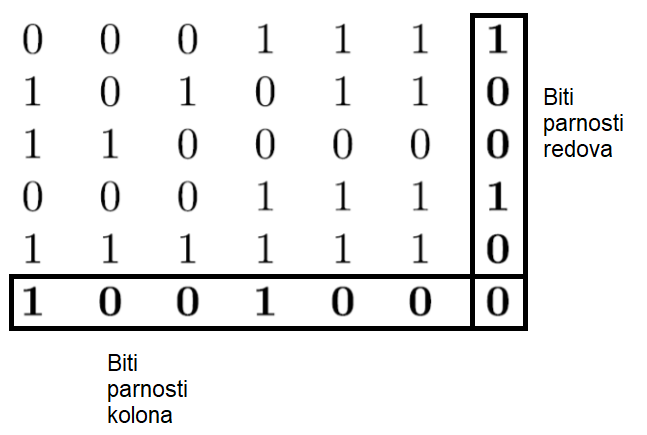
|  |  |  |  |
| --- | --- | --- | --- |
| Instrukcija | Kodovanje | Operacija | Komentar |
| IN A | 011 xxxxxxx | A<-Input | Na ulaz A se dovodi vrednost sa Input |
| OUT A | 100 xxxxxxx | Output<- A | Čita se vrednost iz A (Akumulatora) |
| DEC A | 101 xxxxxxx | A<-A-1 | Umanje se vrednost A za jedan i ponovo šalje na ulaz A |
| JNZ address | 110 xaaaa | if(A!=0) then PC=aaaa | Ako je vrednost A  različita od nule tada PC dobija vrednost aaaa |
| HALT | 111 xxxxx | Halt | Zasutavljanje |

* 1. Overlaping parity

Je koncept koji omogućava da se detektuje i ukloni greška jednog bita.

Overlaping parity se zasniva na tome da jedan bit parnosti pokrije određeni broj bita podataka.

Matrica (Slika 3) podataka veličine 6x5 može viti pokrivena sa 5 bita parnosti redova, 6 bita parnosti kolona i jednim zajedničkim bitom parnosti. Ako dogodi greška na jednom bitu unutar matrice, to će dovesti do toga da se biti parnosti koji pokrivaju taj bit podataka promene.



Slika 3: Overlapping matrica

U primeru sa slike 3 jedan bit podataka je pokriven sa više od jednog bita parnosti. Kako bi se dobilo na efikasnosti, potrebno je odrediti minimalan broj bita parnosti koji pokriva sve bite podataka, to jest omogućava da se otrkije jednobitna greška. To je moguće odrediti koristeći formulu: Gde je r broj bita parnosti, a d broj bita podataka. Primer na kome je primenjen ovaj koncept je poznat pod nazivom (7,4) Hamming single error correcting (SEC). Broj sedam označava dužinu kodne reč, dok broj četiri označava broj bita podataka. Kada se dužina kodne reči oduzme od broja bita podataka dobija se broj potrebnih bita parnosti.

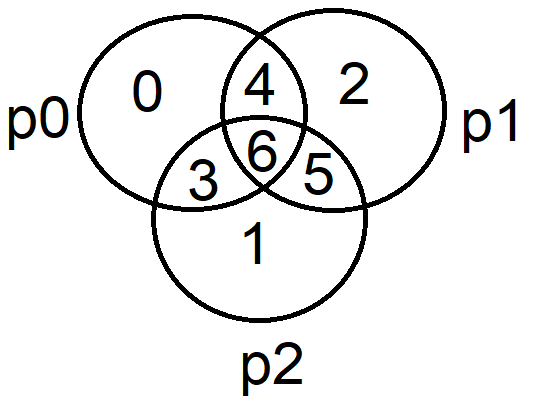
Za r = 3

Kao što se može videti za četiri bita podataka je dovoljno koristiti tri bita parnosti. Tako da koda reč za (7,4) Hamming SEC izgleda: a3a2a1a0p2p1p0 , gde a3a2a1a0 predstavljaju bite podataka, dok p2p1p0 predstavljaju bite parnosti.

Jednobitna greška je predstavljena kao stanje ( Tabela 2). Sindrom je rezultat koji se dobija kao razlika između bita parnosti kodne reči i bita parnosti kodne reči koja sadrži grešku. Tako da sindrom određuje poziciju, to jest bit koji je doveo do greške.

Tabela 2: Raspoređivanje bita parnosti

|  |  |  |
| --- | --- | --- |
| State | Erroneous parity check(s) | Syndrome |
| No errors | None | 000 |
| Bit 0 (p0) error | p0 | 001 |
| Bit 1(p1) error | p1 | 010 |
| Bit 2 (p2) error | p1 | 100 |
| Bit 3 (a0) error | p0,p1 | 011 |
| Bit 4 (a1) error | p0,p2 | 101 |
| Bit 5 (a2) error | p1,p2 | 110 |
| Bit 6 (a3) error | p0,p1,p2 | 111 |



Slika 5: Pokrivenost pozicija

Na osnovu tabele 2 i slike 5 se dobija račun bita parnosti:

p0 = a0 xor a1 xor a3.

p1 = a0 xor a2 xor a3

p2 = a1 xor a2 xor a3

Primer:

Biti podataka su: 1111(a3a2a1a0), tako da će p2 p1 p0 biti 111 respektivno. Tako da je kodna reč 1111111.

Dogodi se greška na a2 tako što se vrednost invertuje. Tada se dobija kodna reč 1011001.

Pošto je a2 bit na kome se dogodila greška, iz tabele 2 se može videti da će p1 p2 biti promenljeni. Sindrom (synd) se računa tako što se na bite parnosti obe reči primeni logička operacija eksluzivno ili (xor): synd = 111 xor 001 = 110

Vrednost sindroam je 110 binarno, to jest 6 dekadno. Dakle na šestoj poziciji se dogodila greška. Ispravljanje te greške je moguće uraditi tako što se izvrši ponovno invertovanje pogrešnog bita.

1. Zaštita memorije

Memorija u kojoj se skladište podaci, to jeste instrukcije je predstavljena kao RAM. Od velike važnosti je da tačni podaci budu upisani u memoriju, tako da je princip otrkivanja I korekcije greške objašnjen u predhodnom poglavlju primenjen na ovu komponentu.

EC1 instrukcije su osmobitne, tako da je prvo potrebno odrediti odgovarajući Hamingov kod.

Neki od standardnih Hamingovih kodova su: Hamming(7,4), Hamming(15,11), Hamming(31,26) (Tabela 3). Kao što je već pomenuto, EC1 ima osam bita podataka, a to nije ni jedan od standardnih Hamingovih kodova. Tako da je potrebno odlučiti koji od standardnih Hamingovih kodova odgovara osmobitnoj magistrali podataka.

U slučaju da se Hamming(15,11) primenjuje na osam podataka, 3 bita podataka bi bilo neiskorišćeno, a broj pariti bita bi bio četiri.

U slučaju da se Hamming(7,4) primenjuje na osam podataka, potrebno je imati dve instance Hamingovog koda. Tako da prva instanca Hamingovog koda predstavlja gornju niblu EC1 magistrale podataka (Gornjih četiri bita), dok bi druga instanca Hamingovog koda predstavljala donju niblu EC1 magistrale podataka (Donjih četiri bita). Broj iskorištenih bita parnosti je šest I na taj način je moguće detektovati dva bita greške sve dok su ti bitovi nalaze u pojedinačnim niblama (Jedan pogrešan bit po nibli).

Tabela 3: Standardni Hamingovi kodovi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parity bits(r) | Total bits (n) | Data bits (k) | Name | Rate |
| 3 | 7 | 4 | Hamming(7,4) | 4/7 = 0.571 |
| 4 | 15 | 11 | Hamming(15,11) | 11/15 = 0.733 |
| 5 | 31 | 26 | Hamming(31,26) | 26/31 = 0.839 |

2.1 Proračun Hamingovog koda (7,4)

Potrebno je izračunati:

1. Generator matrix G = [IkP] => k = 4 biti podataka

Basis vektori su:

v0 = 0 1 1 0 0 1 1 (IN EC1 instrukcija 4 bita (6:3), biti parnosti (2:0))

v1 = 1 0 0 0 1 1 1 (OUT A EC1 instrukcija 4 bita (6:3), biti parnosti (2:0))

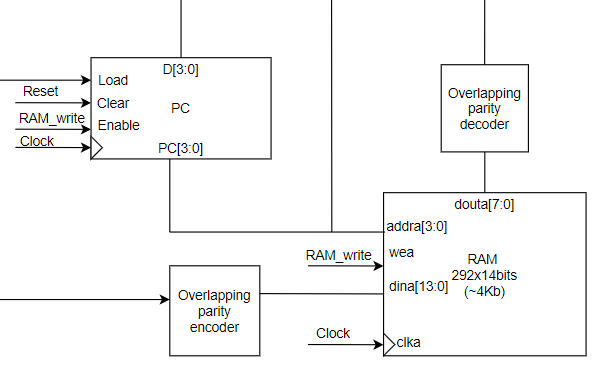
v2 = 1 0 1 1 0 1 0 (DEC A EC1 instrukcija 4 bita (6:3), biti parnosti (2:0))

1. Parity check matrix H = [PT In-k+1]

Primer: Kodna reč c = 0 0 0 0 0 0 0 u slučaju da se tokom prijema signala dogodi greška

r = 0 0 0 0 0 0 1 tada je moguće izračunati sindrom.

S obzirom da je sindrom s = 001 to znači da se greška desila na prvoj poziciji tokom prijema podatka.

1. Implementacija Overlapping parity. 

Slika 6: RAM sa Overlapping parity

