

Građa računala

ISHOD 1

1. [_M, 3 boda] Kod dizajna ALU što se oblikuje prvo: aritmetički ili logički dio (1 bod) i zašto (2 boda)?

Prvo se oblikuje aritmetika, jer su logički sklopovi su sadržani u aritmetici.

2. [_M, 2 boda] Zapišite brojku 3FFA u little endian (1 bod) i u big endian (1 bod) zapisu.

3FFA je big endian zapis, FA3F je little endian.

3. [_M, 2 boda] Demonstrirajte 8 bitno zbrajanje bez prijenosa.

LDA #\$FA

CLC

ADC #\$1

4. [_M, 2 bod] Imamo dvije hipotetske ALU, zadane tablicama stanja. Jedinica A ima 16 mogućih operacija. Jedinica B ima 8. Ako pretpostavimo da je logički dio obje jedinice identičan koja od njih će nužno imati više sklopova i osnovnih logičkih elemenata?

A, jer ima više mogućih operacija.

5. [_Ž, 2 boda] Demonstrirajte overflow s predznakom (signed overflow).

Kada zbrajamo iznad limita adresnog prostora. Npr. ako zbrajamo (signed,2K) brojeve 7 (0111) i 4 (0100) desit će se overflow u bit predznaka.

6. [_Ž, 3 boda] Objasnite na primjerima koje su uloge kontrolne jedinice (CU)?

Uloga kontrolne jedinice je kao uloga prometnog policajca. Regulira sve što procesor radi upravljajući ostalim komponentama.

7. [_M, 4 boda] Što se kolokvijalno naziva FE (Front End) (1 bod), a što BE (Back End) (1 bod) kod procesora i koje zadaće izvršavaju (2 boda)?

Front end -> fetch, decode.

Back and dođe i preuzme instrukciju koja mu je pripremljene u dekođeru te je krene izvršavat. Sastoji se od executea.

ISHOD 2

1. [_M, 3 boda] Korištenjem 6502 assemblera popunite svaku parnu lokaciju između \$200 i \$2FF sa vrijednosti decimalno 13.

LDA #\$0D

LOOP:

STA \$200,X

INX

INX

BNE LOOP

2.[_M, 2 boda] Prethodni zadatak proširite tako da popunjava i lokacije između \$300 i \$3FF ali popunite neparne lokacije.

LDA #\$0D

LOOP:

STA \$301,X

INX

INX

BNE LOOP

3. [_M, 1 bod] Što je vremenski brže - pristup registru ili pristup memoriji (0.5 bodova) i zašto (0.5 bodova)?

Pristup registru, jer je već brže samim time što je fizički u procesoru te je time i direktno povezan na sabirnice i nema potrebe za pripremom kao kod memorije.

4. [_Ž, 3 boda] Kako branch predictor (sklop koji služi za predviđanje grananja) pomaže brzini izvršavanja instrukcija?

Branch predictor je proces koji odlučuje koje instrukcije treba preloadati u procesor. Branch predictor čita memoriju i predviđa koje će instrukcije biti sljedeće u izvršavanju.

5. [_Ž, 2 boda] Izračunajte prostornu (1 bod) i vremensku (1 bod) složenost oba rješenja iz prethodnih zadataka (količinu zauzete memorije i broj ciklusa za cijelo izvršavanje).

Budući da su oba zadatka jednako složena imaju istu vremensku i prostornu složenost koja je:

- Vremenska: $2 + (5+2+2)*256 - 1$
- Prostorna: 9 bajtova

ISHOD 3

1. [_M, 3 boda] Na lokacijama \$400 i \$410 nalaze se tri bajta koja sadržavaju redom: praznu lokaciju i 16 bitni broj u little endian zapisu. Izračunajte zbroj ova dva broja i zapišite ga na lokaciju \$420 koristeći registar Y za spremanje potencijalnog prijenosa (carry)

//nek neko ovo napravi, Jasmin je prico da je u task 4 nesto helpful

2. [_M, 3 boda] Kakve sve vrste instrukcija poznaje kontrolna jedinica (1 bod) i po čemu se te vrste instrukcija razlikuju (2 boda)?

Time što je definirano u instruction setu. Razlikuje ih po tome je li mijenjaju ili ne mijenjaju flow.

3. [_M, 3 boda] Predložite - ma koliko potencijalno bili nemogući - barem dva, što jednostavnija načina za rješavanje RowHammer problema.

Usporiti memoriju ukoliko se češće flippaju bitovi kako bi bila stabilnija. Razmaknuti ćelije ili povećati veličinu memorije. Također napredne metode poput PARA-e itd. Iako trenutno idealno rješenje ne postoji.

4. [_M, 1 bod] Koja je razlika između ranjivosti Spectre i RowHammer?

Spectre radi na osnovu spekulativnog izvršavanja te timingom se provjerava je li procesor pristupio nečemu što nije smio što nam govori je li izvršio nešto što nije trebao izvršiti. Meltdown radi tako da provjerava je li podatak u cacheu ili ne ovisno o vremenu potrebnom da to napravi.

5. [_Ž, 3 boda] Zadatak 1 rješite korištenjem subroutine, lokaciju prvog broja rutina mora pročitati sa \$10, \$11, drugi i treći broj se nalazi na lokacijama +\$10 i +\$20. Primjerice ako je na lokaciji \$10 zapisano 00 a na lokaciji \$11 vrijednost \$03, prvo broj koji rutina mora zbrojiti nalazi se stvarno na lokaciji \$300, drugi na \$310 a rezultat na \$320.

6. [_Ž, 2 boda] Koje centralne procesne jedinice (CPU) ne "boluju" od Meltdown ranjivosti (1 bod) i zašto (1 bod)?

AMDovi procesori ne boluju od Meltdown ranjivosti jer AMD radi na drugi princip te štiti memoriju.