# Primer 1

Za datu kodnu sekvencu izlistati sve zavisnosti-po-podacima (*data dependencies*) tipa RAW, WAR i WAW.

I0: A = B + C;

I1: C = A – B;

I2: D = A + C;

I3: A = B \* C \* D;

I4: C = F / D;

I5: F = A - G;

I6: G = F + D;

## Rešenje:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RAW zavisnost | | WAR zavisnost | | WAW zavisnost | |
| od instrukcije | ka instrukciji | od instrukcije | ka instrukciji | od instrukcije | ka instrukciji |
| I0 | I1 | I0 | I1 | I0 | I3 |
| I0 | I2 | I1 | I3 | I1 | I4 |
| I1 | I2 | I2 | I4 |  |  |
| I3 | I5 | I3 | I4 |  |  |
| I2 | I3 | I4 | I5 |  |  |
| I1 | I3 | I5 | I6 |  |  |
| I2 | I4 |  |  |  |  |
| I3 | I5 |  |  |  |  |
| I5 | I6 |  |  |  |  |

# Primer 2

U sledećoj programskoj sekvenci identifikovati

* RAW hazarde,
* WAR hararde,
* WAW hazarde
* Upravljačke (*control*) hazarde

1. Div r2, r5, r8

2. Sub r9, r2, r7

3. Ash r5, r14, r6

4. Mul r11, r9, r5

5. Beq r10, #0, r12

6. Or r8, r15, r2

## Rešenje:

* RAW hazardi postoje izmedju sledećih instrukcija:

1-2; 3-4; 2-4; 1-6

* WAR hazardi postoje izmedju sledećih instrukcija:

1-3; 1-6

* Hazardi tipa WAW ne postoje u ovoj sekvenci:
* U sekvenci postoji samo jedan *control*-hazard, a on je izmedju instrukcija:

5-6

# Primer 3

Korišćenjem tehnike zakašnjenog grananja, smanjiti zastoje u datom programu.

Polazni program:

Loop: LD F0, 0(R1)

ADD F4, F0, F2 SD 0(R1), F4

SUBI R1, R1, #8

BNEZ R1, Loop

## Rešenje:

I varijanta, u prozor zakašnjenog grananja se postavlja instrukcija sa ciljne adrese grananja, tj. instrukcija na labeli Loop

LD F0, 0(R1)

Loop: ADD F4, F0, F2

SD 0(R1), F4

SUBI R1, R1, #8

BNEZ R1, Loop

LD F0, 0(R1) (instrukcija postavljena u delay slot)

II Varijanta: Pošto u originalnom program SUBI dekrementira sadržaj R1 za 8, a u preuređenom program se SD poslavlja nakon SUBI, potrebno je modifikovati adresu na koju SD pamti

Loop: LD F0, 0(R1)

ADD F4, F0, F2

SUBI R1, R1, #8

BNEZ R1, Loop

SD 8(R1), F4 (instrukcija postavljena u delay slot)

# Primer 4

* *Zakašnjeni load i preuređenje instrukcija da bi se izbegli zastoji*

EX faza za instrukcije ADD i SUBD traje 2 ciklusa (instrukcije rade sa podacima u pokretnom zarezu). Pretpostaviti da se koristi premošćavanje. Preurediti program tako da se broj zastoja minimizuje.

Polazni program:

LD F0, 0(R1)

ADD F4, F0, F2

SD F4, 0(R1)

LD F6, -8(R1)

SUBD F8, F6, F2

SD F8, -8(R1)

## Rešenje

Preuređeni program bez zastoja:

LD F0, 0(R1)

LD F6, -8(R1)

ADD F4, F0, F2

SUBD F8, F6, F2

SD F4, 0(R1)

SD F8, -8(R1)

# Primer 5

Neka je data sledeća sekvenca aktuelnih ishoda za *single-static Branch*:

**TTTNTNTTTNTNTTTNTN**

T znači da je došlo do grananja (*branch taken*), a N do grananja nije došlo (*branch not taken*). U konkretnom slučaju pretpostaviti da je to jedinstvena instrukcija tipa *Branch* u programu.

Usvojiti da se predviđanje ishoda grananja obavlja korišćenjem jedno-bitnih brojača. Brojači se inicijaliziraju na stanje N. Koja će od grananja (*branches*) u datoj sekvenci biti pogrešno predviđena (*miss-predicted*)?

## Rešenje

|  |  |  |
| --- | --- | --- |
| stanje prediktora pre predikcije | ishod grananja | pogrešna predikcija? |
| N | T | Da |
| T | T | Ne |
| T | T | Ne |
| T | N | Da |
| N | T | Da |
| T | N | Da |
| N | T | Da |
| T | T | Ne |
| T | T | Ne |
| T | N | Da |
| N | T | Da |
| T | N | Da |
| N | T | Da |
| T | T | Ne |
| T | T | Ne |
| T | N | Da |
| N | T | Da |
| T | N | Da |

# Primer 6

Program P sadrži dve naredbe uslovnog grananja CB-1 i CB-2. Ishodi grananja zavise od vrednosti promenljive A u programu. U sledećoj tabeli dati su ishodi naredbi grananja kada se program P izvršava za različite vrednosti promenljive A (T – grananje se obavlja, NT – grananje se ne obavlja)

|  |  |  |
| --- | --- | --- |
| Vrednost A | CB-1 | CB-2 |
| 1 | T | T |
| 2 | NT | NT |
| 3 | NT | T |

1. Ako se koriste 1-bitni prediktori koji su inicijalizovani na “NT” popuniti tabelu pri čemu CB-P označava predikciju, CB-A akciju, a CB-U novu vrednost prediktora

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | CB-1:P | CB-1:A | CB-1:U | CB-2:P | CB-2:A | CB-2:U |
| 2 |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

1. Ako se koristi 2-bitni predictori koji su implementirani kao zasićeni brojači koji su inicijalizovani na 00, popuniti i tabelu pri čemu CB-P označava predikciju, CB-A akciju, a CB-U novu vrednost prediktora

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | CB-1:P | CB-1:A | CB-1:U | CB-2:P | CB-2:A | CB-2:U |
| 2 |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |