Assignment 4 write up

Q1

Latency of accesses is found by using time stamp counter(rtdscp)

We will find the value in time stamp before the process and after the process.

Difference is taken between these start and end values of time stamp counter

Cpuid is used in order to execute the the instructions in process in sequential order

Q2

The actual specifications of the L1 cache are:

L1\_I CACHE size -65536

L1\_I CACHE size -4

L1\_I CACHE size -64

L1\_D CACHE size -32768

L1\_D CACHE size -8

L1\_D CACHE size -64

The command we used to find the specifications is “getconf -a | grep CACHE”

Q3

(i)

Step 1: We will create an array of size 256.

Step 2: We will empty the cache blocks which have the address of the array elements of the array we had initialised so that they will not interfere while measuring the clock cycles.

Step 3: We will access the array elements one after the other from index 0 to 255.for the first access of array element it will bring the address of a[0] to a[block\_size-1] to cache block and we go on accessing the elements with increasing index by 1 when we access a[block\_size] there will be a cache miss and we can see increase in the hit number of CPU cycles.

Step 4:We will take an average of 32 such process and compute average no of CPU cycles

(ii)The output is:

Index no of CPU cycles

57 0

56 1

56 2

52 3

334 4

55 5

56 6

53 7

52 8

53 9

54 10

55 11

56 12

53 13

56 14

57 15

56 16

55 17

56 18

69 19

329 20

55 21

53 22

53 23

57 24

55 25

53 26

60 27

53 28

56 29

53 30

56 31

57 32

56 33

55 34

57 35

171 36

56 37

54 38

56 39

57 40

56 41

56 42

54 43

54 44

55 45

58 46

54 47

58 48

56 49

61 50

57 51

288 52

55 53

57 54

54 55

58 56

58 57

56 58

57 59

56 60

56 61

55 62

53 63

56 64

53 65

57 66

56 67

196 68

56 69

53 70

51 71

56 72

54 73

55 74

54 75

56 76

57 77

55 78

56 79

56 80

56 81

56 82

53 83

148 84

56 85

55 86

54 87

57 88

56 89

55 90

53 91

55 92

53 93

56 94

56 95

57 96

58 97

56 98

57 99

156 100

53 101

53 102

54 103

50 104

56 105

56 106

56 107

54 108

57 109

58 110

55 111

53 112

55 113

55 114

53 115

138 116

56 117

55 118

52 119

54 120

55 121

57 122

53 123

53 124

56 125

54 126

56 127

52 128

56 129

53 130

53 131

141 132

56 133

58 134

54 135

54 136

56 137

57 138

56 139

58 140

56 141

54 142

53 143

55 144

55 145

53 146

54 147

191 148

53 149

58 150

53 151

56 152

58 153

58 154

55 155

56 156

55 157

56 158

53 159

52 160

57 161

54 162

55 163

222 164

55 165

54 166

54 167

53 168

56 169

58 170

56 171

56 172

53 173

51 174

53 175

56 176

56 177

54 178

58 179

197 180

52 181

52 182

56 183

56 184

54 185

55 186

56 187

56 188

53 189

56 190

56 191

54 192

56 193

54 194

54 195

204 196

54 197

57 198

55 199

54 200

54 201

55 202

55 203

56 204

52 205

55 206

55 207

55 208

58 209

56 210

56 211

141 212

53 213

53 214

53 215

53 216

53 217

53 218

54 219

53 220

53 221

56 222

56 223

54 224

58 225

56 226

53 227

186 228

54 229

55 230

56 231

53 232

55 233

59 234

54 235

54 236

52 237

52 238

53 239

53 240

52 241

56 242

56 243

56 244

55 245

56 246

60 247

56 248

56 249

57 250

53 251

54 252

53 253

56 254

54 255

(iii)Yes, it did match the cache block size of the cache.

We can observe that we can see that there is an increase in no of CPU cycles for every 16 memory accesses . We know that the array elements are int type so they take 4 bits of data. So there a cache miss for every 4\*16 = 64 bits hence that cache block size 64

Q4)

(i)

Step 1:As we know the size of catch is 64 and we know the number of sets

Step 2: Using this we can access adjacent blocks in a set(difference between 2 adjacent blocks )

difference = number of sets \* block size

(ii)access times of memorys accessed earlier after accessing a new block in cache

69

92

69

69

69

69

69

69

69

92

69

69

69

115

92

69

115

115

69

115

92

69

115

69

69

69

92

138

69

115

115

115

115

69

115

115

69

115

115

230 – cache miss when we access for the 9th block in a set as the 8 block are filled one of the old accessed data gets evicted to store new data. So the associativity is 8.

115

92

115

92

69

92

92

115

207

115

115

115

69

115

115

(iii)yes, we got that the associativity as 8

Which is compatible with our model.

This summaraises the write up report of CS20B063 and CS20B058. Thank you