Final CS 3339

A screenshot of a cell phone

Description automatically generated

-The Temporal reference for locality: reusing of same element within short period of time

- Spacial locality of reference: Reusing of data elements which are nearby to original location

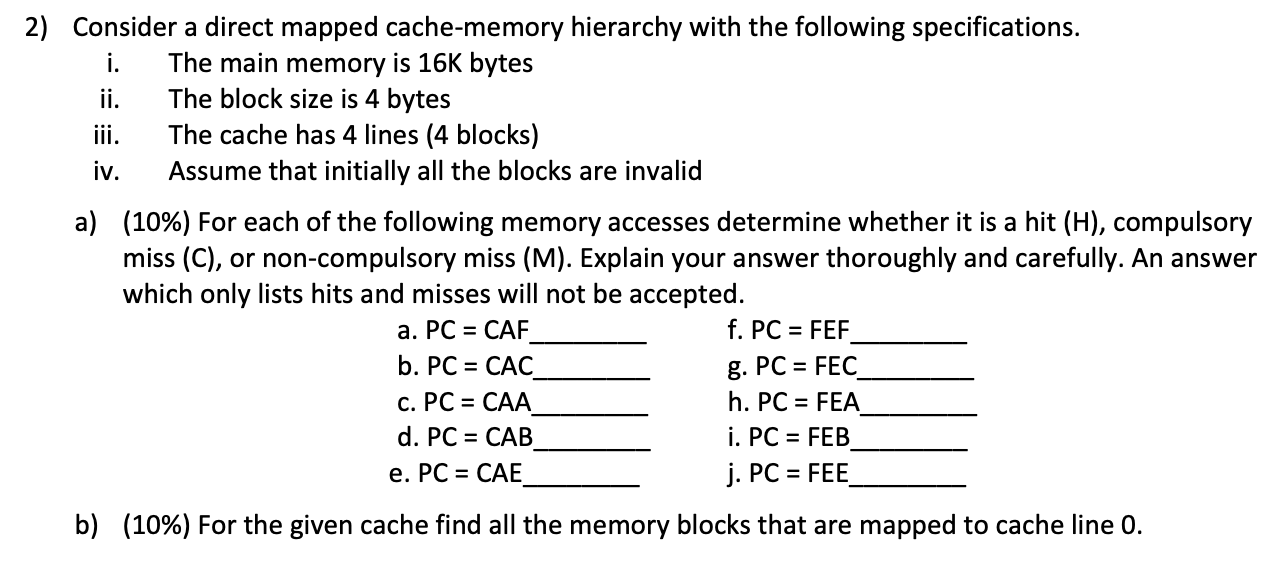
1) Matrix C has spacial locality of refrence as C[i][j]

-we access adjacent element in column frequently because loop of j runs from 0 to N.

2) Matrix B has spacial locality of reference as for a fixed value of K B[k][j]

-Access to n adjacent subsequent locations in the same row

3) Matrix A has no locality as , A[i][k] will always have more than n interleaved location each time we access it, No temporal locality as A[i][k] will never repeat itself for access



The main memory is 16K bytes

CPU’s physical address = log2 16K = 14 bit

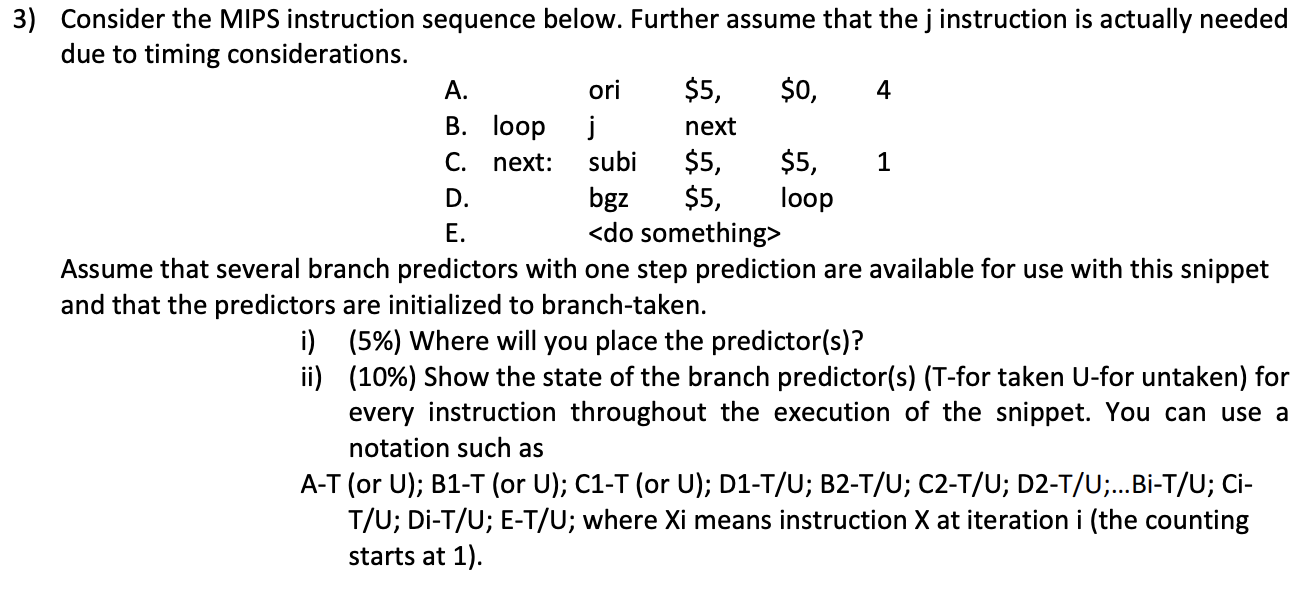
4 blocks x 4 bytes = 2^4, therefore 4 bits

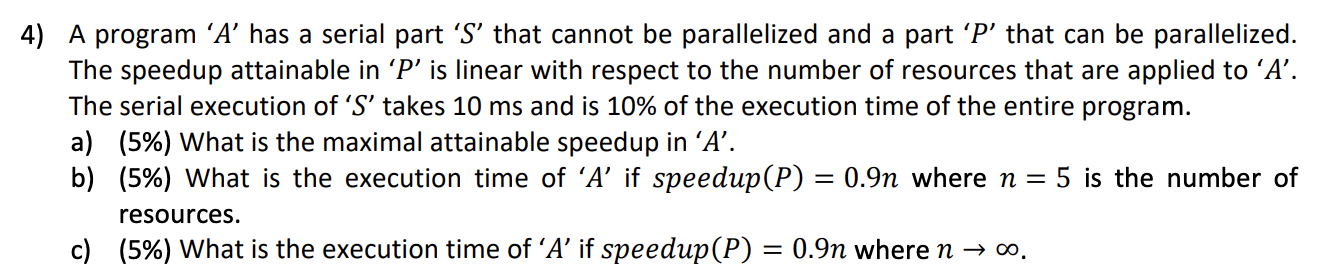
Block offset = log2 4 = 2 bit

Cache has 4 lines = log2 4 = 2 bits

Tag bit = 14 – 2 – 2 = 10 bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PC = CAF | 1100 1010 1111 | Tag(0011001010)  Line(11)  Offset (11) | C | The value was brought into cache and the cache is empty |
| PC = CAC | 1100 1010 1100 | Tag(0011001010)  Line(11)  Offset (00) | H | Line went to the 3rd line of cache , tag bits matches with tag bits of CAF |
| PC = CAA | 1100 1010 1010 | Tag(0011001010)  Line(10)  Offset (10) | C | The value was brought into cache and goes to 2nd line |
| PC = CAB | 1100 1010 1011 | Tag(0011001010)  Line(10)  Offset (11) | H | Line went to the 2rd line of cache , tag bits matches with tag bits of CAA |
| PC = CAE | 1100 1010 1110 | Tag(0011001010)  Line(11)  Offset (10) | H | Line went to the 3rd line of cache , tag bits matches with tag bits of CAC |
| PC = FEF | 1111 1110 1111 | Tag(0011111110)  Line(11)  Offset (11) | C | Line went to the 3rd line of cache , tag bits not matching |
| PC = FEC | 1111 1110 1100 | Tag(0011111110)  Line(11)  Offset (00) | C | tag bits not matching |
| PC = FEA | 1111 1110 1010 | Tag(0011111110)  Line(10)  Offset (10) | H | Tag bits matches with FEA |
| PC = FEB | 1111 1110 1011 | Tag(0011111110)  Line(10)  Offset (11) | H | Line goes to 2nd line, tag bits matches with FEA |
| PC = FEE | 1111 1110 1110 | Tag(0011111110)  Line(11)  Offset (10) | H | Line goes to 3nd line, tag bits matches with FEF |

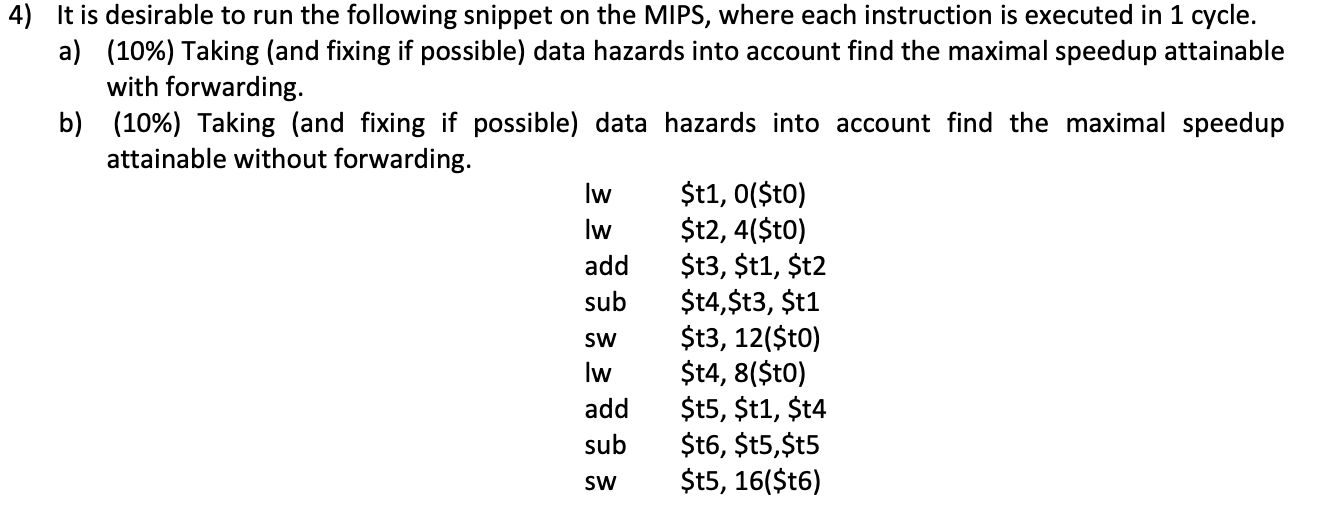




(a)execution time of S = 10 ms  
execution time of A = 10 \* 10 = 100ms  
speedup will be maximum when execution time of P = 0  
maximum speedup = 100/(10 + 0) = 10

(b)speedup(P) = 0.9n  
here n = 5  
speedup(P) = 0.9\*5 = 4.5  
execution time of P = 90/4.5 = 20 ms  
execution time of A = 10 + 20 = 30 ms

(c)speedup(P) = 0.9n  
here n = infinite  
speedup(P) = 0.9\*infinite = infinite  
execution time of P = 90/infinite = 0 ms  
execution time of A = 10 + 0 = 10 ms



I1: t1 <- M [0 + t0]

I2: t2 <- M [4 + t0]

I3: t3 <- t1 + t2

I4: t4 <- t3 – t1

I5: t3 -> M [12 + t0]

I6: t4 <- M [8 + t0]

I7: t5 <- t1 + t4

I8: t6 <- t5 – t5

I9: M [16 + 16] <- t5

Raw hazard and data dependency

[I3 – I2] for t2

[I4 – I3] for t3

[I7 – I8] for t4

[I8 – I7] for t5

Number of cycles taken 9 \* 5 = 45 cycles

Forwarding: 14 cycles

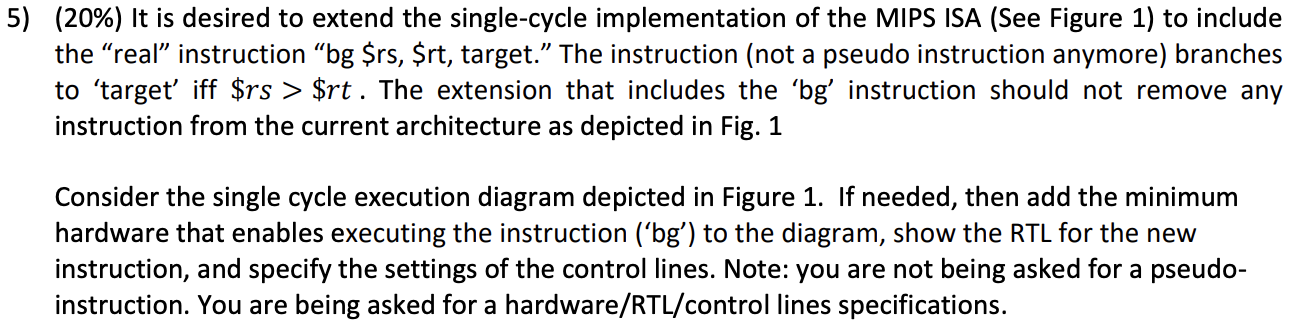
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| I1 | IF | ID | EX | MA | WB |  |  |  |  |  |  |  |  |  |
| I2 |  | IF | ID | EX | MA | WB |  |  |  |  |  |  |  |  |
| I3 |  |  | IF | ID | ID | EX | MA | WB |  |  |  |  |  |  |
| I4 |  |  |  | IF | - | ID | EX | MA | WB |  |  |  |  |  |
| I5 |  |  |  |  |  | IF | ID | EX | MA | WB |  |  |  |  |
| I6 |  |  |  |  |  |  | IF | ID | EX | MA | WB |  |  |  |
| I7 |  |  |  |  |  |  |  | IF | ID | EX | MA | WB |  |  |
| I8 |  |  |  |  |  |  |  |  | IF | ID | EX | MA | WB |  |
| I9 |  |  |  |  |  |  |  |  |  | IF | ID | EX | MA | WB |

Speed up = 45/14 = 3.21 (3.21 times faster than non-pipeline)

Without Forwarding:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 20 |  |
| I1 | IF | ID | EX | MA | WB |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I2 |  | IF | ID | EX | MA | WB |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I3 |  |  | IF | ID | - | - | EX | MA | WB |  |  |  |  |  |  |  |  |  |  |  |
| I4 |  |  |  | IF | - | - | ID | - | - | ID | EX | WB |  |  |  |  |  |  |  |  |
| I5 |  |  |  |  |  |  | IF | - | - | ID | EX | MA | WB |  |  |  |  |  |  |  |
| I6 |  |  |  |  |  |  |  |  |  | IF | ID | EX | MA | WB |  |  |  |  |  |  |
| I7 |  |  |  |  |  |  |  |  |  |  | IF | ID | - | - | EX | MA | WB |  |  |  |
| I8 |  |  |  |  |  |  |  |  |  |  |  | IF | - | - | ID | - | - | EX | MA | WB |
| I9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | IF | - | - | ID | EX |  |

Speed up = 45/21 = 2.12 (only 2.14 times faster)



A close up of a map

Description automatically generated

bg $rs, $rt,

$𝑟𝑠 > $𝑟t

RTL Jr is an R-type instruction

Pc <- r[rs]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inst | ALU Src | Reg Dst | Reg write | Mem to Reg | Mem Write | Mem  Read | Alu Op | Brach | jump |
| jr | x | x | 0 | x | 0 | x | xx | 0 | 1 |