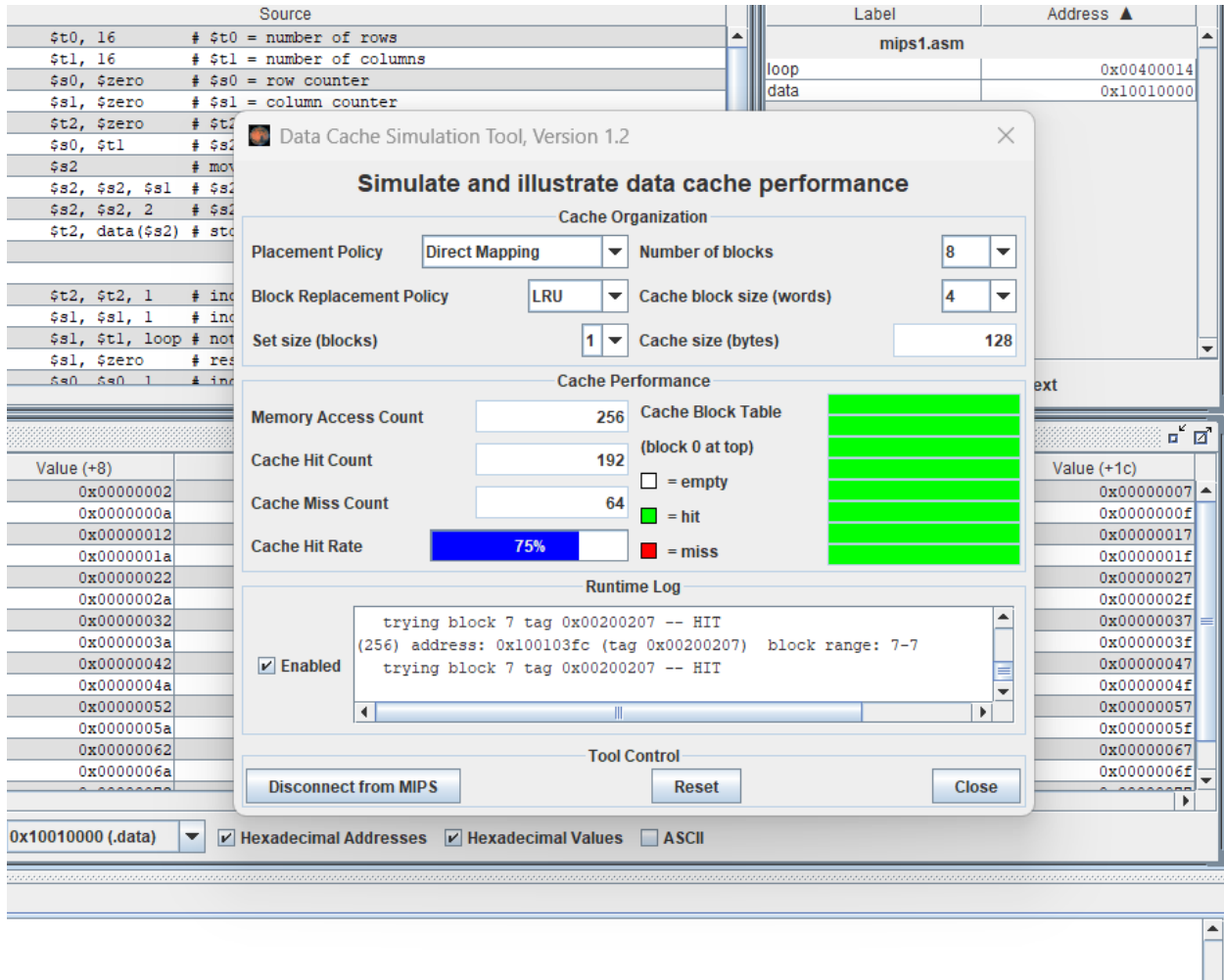


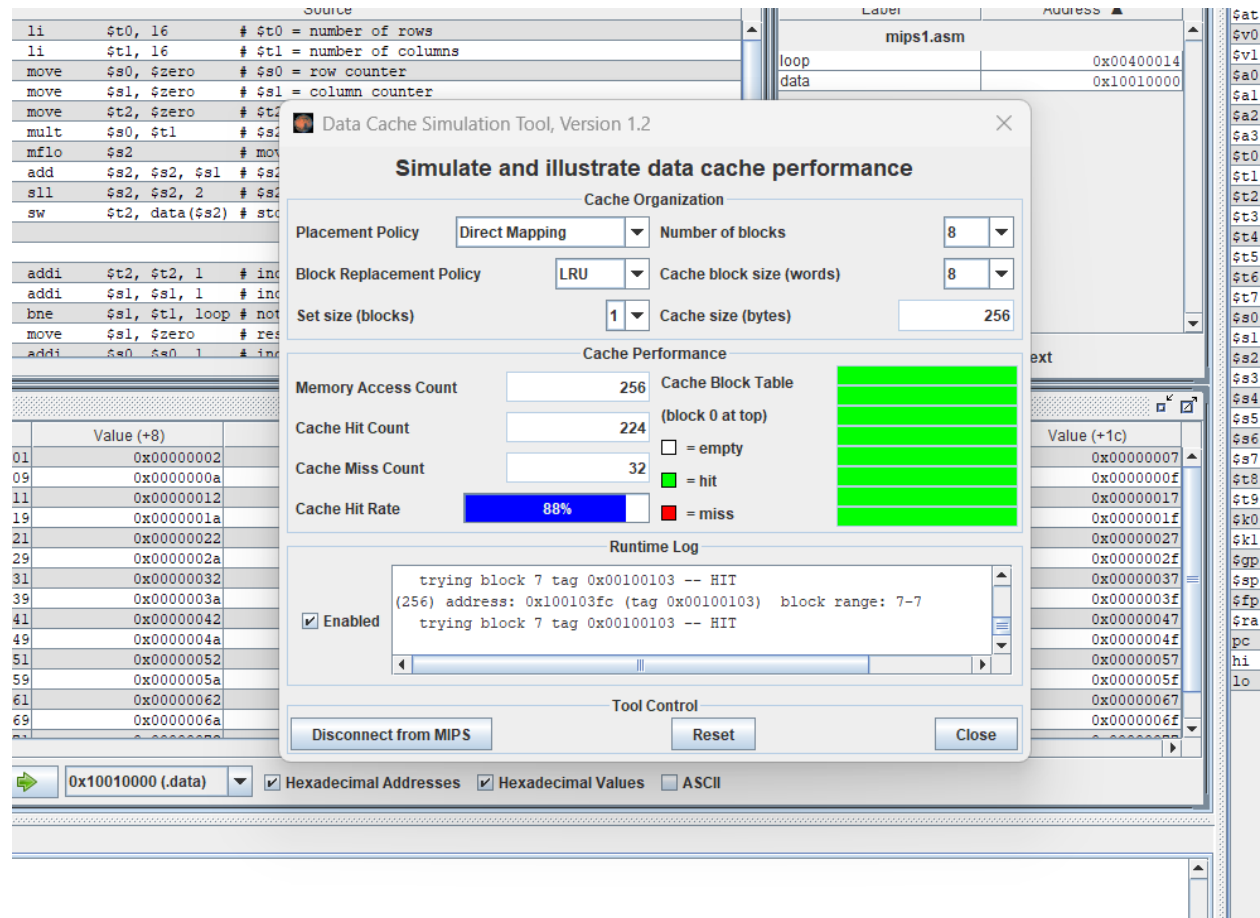
Báo cáo Lab 12

Vũ Hoàng Việt – 20225777

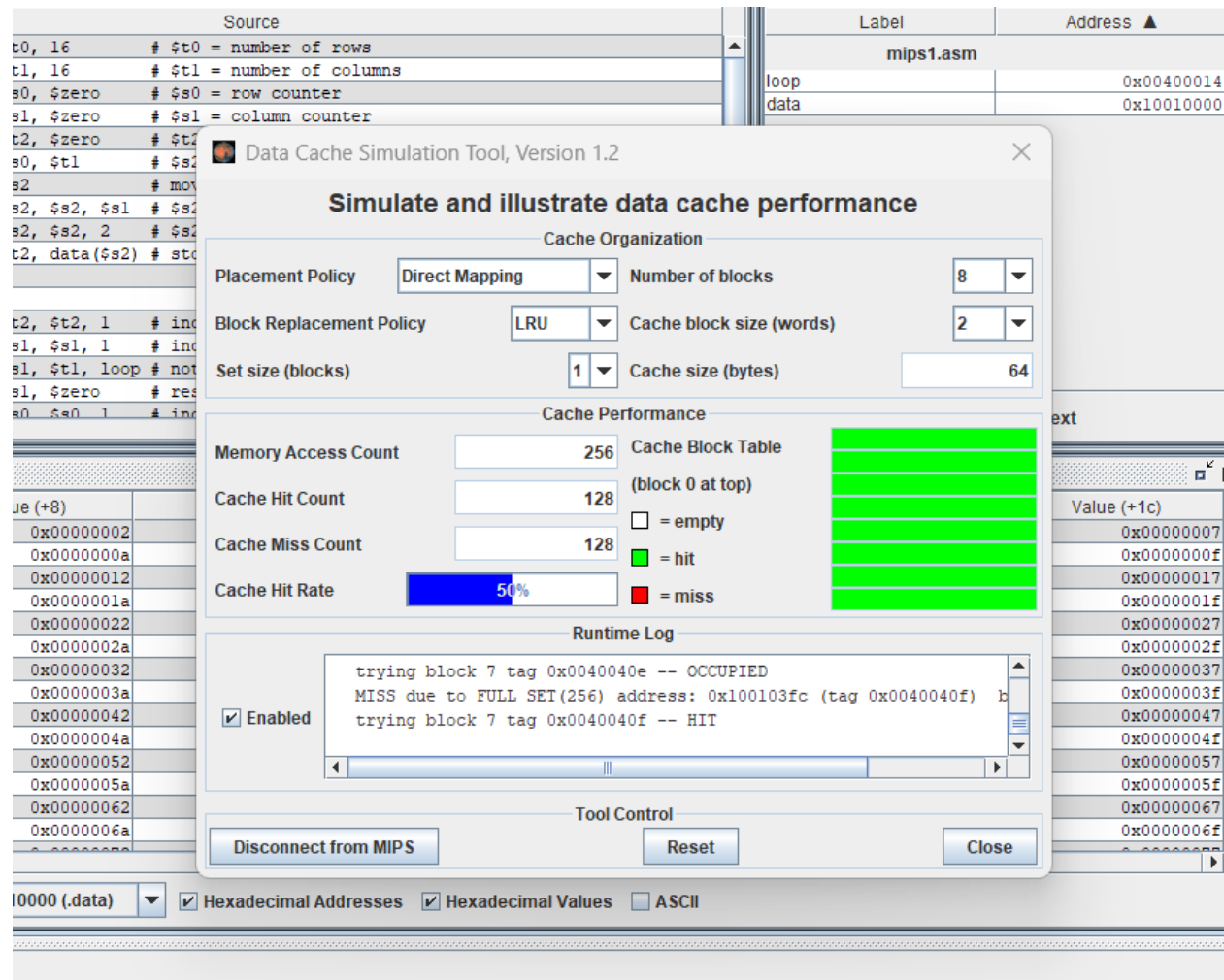
Assignment 1:



- Cache Hit Rate là 75%. Do mỗi lần Cache miss xảy ra, 1 khối gồm 4 kí tự được lấy vào trong cache. Đoạn code truy cập các chỉ số của mảng theo thứ tự mà các phần tử được lưu trong bộ nhớ. Nên với mỗi lần cache miss xảy ra, lấy 1 block về thì 3 lần sau ta sẽ có cache hit (do phần tử ở cùng 1 block với phần tử bị cache miss). Vì vậy tỉ lệ sẽ là $\frac{3}{4}$ (75%)
- Tương tự như cách giải thích trên, khi kích thước khối là 8, ta sẽ có tỉ lệ Cache Hit là 87.5% (xấp xỉ 7/8). Khi kích thước khối là 2 thì tỉ lệ Cache Hit là 50% (1/2).
- Thử chạy trên máy:
 - o Kích thước khối : 8

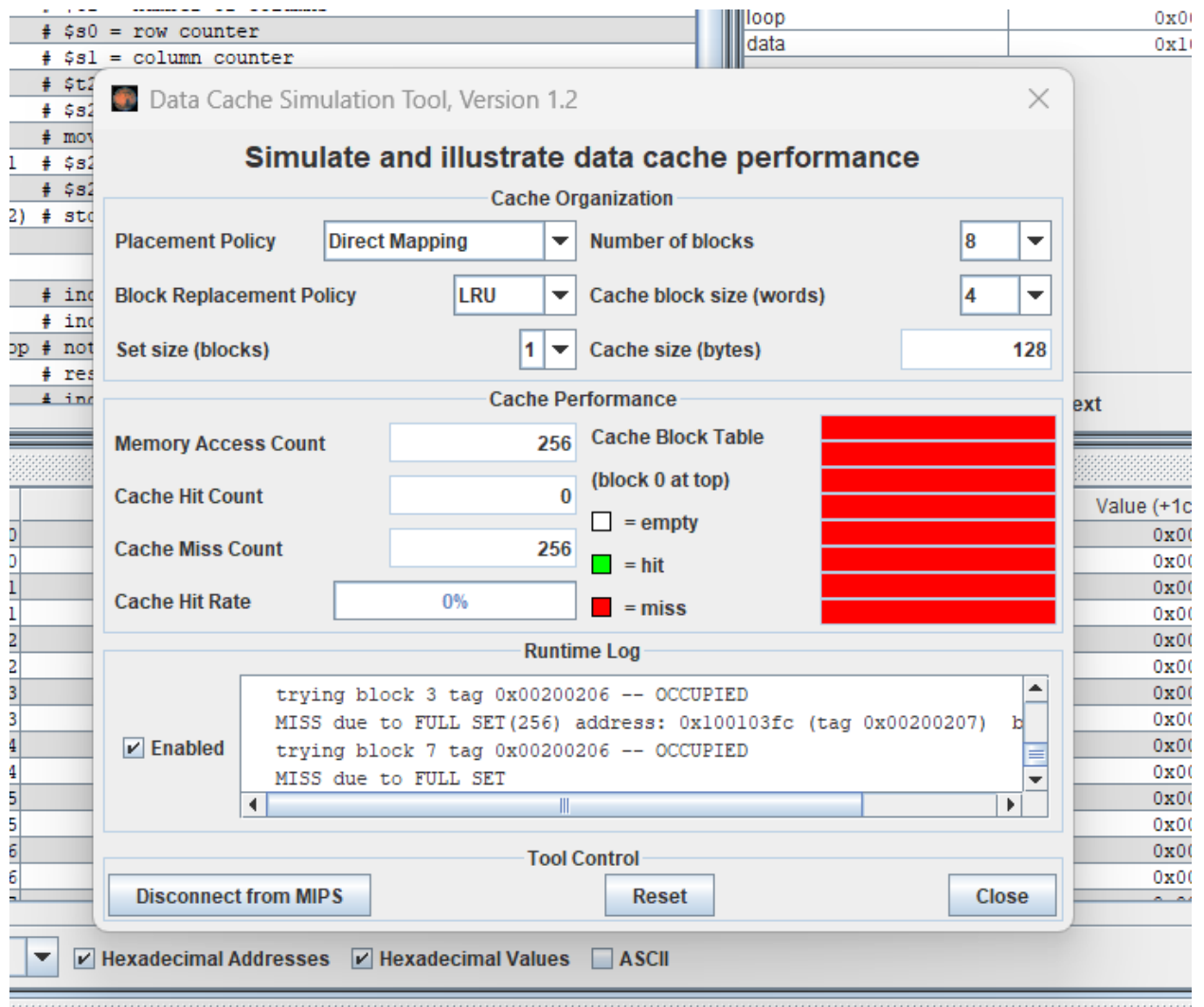


- Kích thước khối: 2

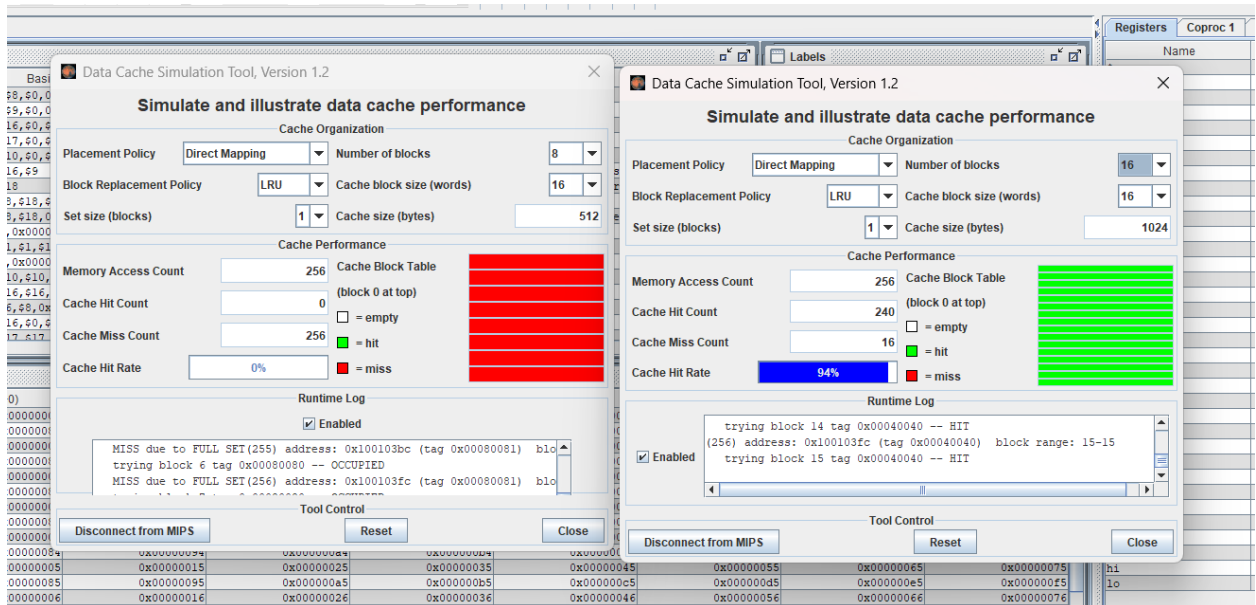


➔ Kết quả đúng với dự đoán

- Chuyển sang chương trình trong file column-major.asm, ta được kết quả:



- Tỷ lệ này xảy ra do chương trình không truy cập tuần tự như trước, mà mỗi phần tử truy cập cách nhau 16 words. Với cách thiết lập của bài (chỉ có 8 blocks trong cache), sẽ không có 2 block nào được truy cập 2 lần. Dẫn tới mỗi lần truy cập phần tử đều là cache miss -> Tỷ lệ cache hit là 0%
- Thay đổi kích thước khối lên 16 và thử chạy với số block là 8 và 16



- Có sự khác biệt lớn về tỉ lệ, do với số block là 8, chương trình vẫn không thể truy cập 2 lần trên một block bất kì, dẫn tới mỗi lần truy cập đều là cache miss -> tỉ lệ cache hit 0%. Với số block là 16, có thể chứa hoàn toàn kích thước của mảng truy cập, nên xảy ra 16 lần cache miss chính là 16 lần đưa block vào cache memory, còn những lần sau thì toàn bộ mảng đều ở trong bộ nhớ nên luôn là cache hit => tỉ lệ cache Hit:

$$(256-16)/256 = 94\%$$

Assignment 2:

- Chương trình row-major.asm

The screenshot displays a MIPS simulator interface. On the left, the assembly code for 'row-major.asm' is shown, including instructions like 'li \$t1, 16', 'move \$s0, \$zero', 'move \$s1, \$zero', 'move \$t2, \$zero', 'loop: mult \$s0, \$t1', 'mflo \$s2', 'add \$s2, \$s2, \$s1', 'sll \$s2, \$s2, 2', and 'sw \$t2, data(\$s2)'. The main window shows a 'Memory Reference Visualization, Version 1.0' dialog box. This dialog has a title 'Visualizing memory reference patterns' and several settings: 'Show unit boundaries (grid marks)' is checked, 'Memory Words per Unit' is 1, 'Unit Width in Pixels' is 16, 'Unit Height in Pixels' is 16, 'Display Width in Pixels' is 256, 'Display Height in Pixels' is 256, and 'Base address for display' is '0x10010000 (static data)'. A 'Counter value 10' is shown with a slider. The visualization area shows a grid of blue and black squares. At the bottom, there are buttons for 'Disconnect from MIPS', 'Reset', 'Help', and 'Close'. Below the dialog, a table shows memory values in hexadecimal.

Value (+4)	Value (+8)
0x00000011	0x00000012
0x00000019	0x0000001a
0x00000021	0x00000022
0x00000029	0x0000002a
0x00000031	0x00000032
0x00000039	0x0000003a
0x00000041	0x00000042
0x00000049	0x0000004a
0x00000051	0x00000052
0x00000059	0x0000005a
0x00000061	0x00000062
0x00000069	0x0000006a
0x00000071	0x00000072
0x00000079	0x0000007a

- Chương trình column-major.asm

