

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



KIẾN TRÚC MÁY TÍNH - CO2008

BÁO CÁO BÀI TẬP LỚN CÁ NHÂN ĐỀ 3

Giảng viên: Nguyễn Xuân Minh

SV thực hiện: Phạm Hoàng Đức Huy – 2011286

Tp. Hồ Chí Minh, Tháng 11/2021



Mục lục

1	Trả lời câu hỏi	2
2	Giải pháp hiện thực	3
3	Giải thuật	3
4	Thống kê số lệnh, loại lệnh sử dụng trong chương trình	5
5	Thời gian thực thi	5
6	Kiểm tra chương trình	5
7	Tổng kết	7
	Tài liệu tham khảo	7

1 Trả lời câu hỏi

Câu 1: Em trình bày bắt đầu từ phần: **2 Giải pháp hiện thực.**

Câu 2:

- a) Bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 1 word:

Địa chỉ word	Địa chỉ theo bit	Tag	Index	Hit/Miss
5	0000 0101	0	5	Miss
174	1010 1110	10	14	Miss
45	0010 1101	2	13	Miss
6	0000 0110	0	6	Miss
253	1111 1101	15	13	Miss
88	0101 1000	5	8	Miss
173	1010 1101	10	13	Miss
14	0000 1110	0	14	Miss
89	0101 1001	5	9	Miss
44	0010 1100	2	12	Miss
186	1011 1010	11	10	Miss
252	1111 1100	15	12	Miss

- b) Bộ nhớ cache Direct-mapped có 8 block, mỗi block chứa 2 word:

Địa chỉ word	Địa chỉ theo bit	Tag	Index	Offset	Hit/Miss
5	0000 0101	0	2	1	Miss
174	1010 1110	10	7	0	Miss
45	0010 1101	2	6	1	Miss
6	0000 0110	0	3	0	Miss
253	1111 1101	15	6	1	Miss
88	0101 1000	5	4	0	Miss
173	1010 1101	10	6	1	Miss
14	0000 1110	0	7	0	Miss
89	0101 1001	5	4	1	Hit
44	0010 1100	2	6	0	Miss
186	1011 1010	11	5	0	Miss
252	1111 1100	15	6	0	Miss

- c) Tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache Direct-mapped với n block tính theo công thức:

$$n \times (block\ size + tag\ size + valid\ field\ size)$$

- Trường hợp câu a: Với 16 block, valid field size là 1 bit, tag size là 28 bits, block size 32 bits (1 word). Tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache là:

$$n \times (block\ size + tag\ size + valid\ field\ size) = 16 \times (32 + 28 + 1) = 976\ bits$$

- Trường hợp câu b: Với 8 block, valid field size là 1 bit, tag size là 28 bits, block size là 64 bits (2 word). Tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache là:

$$n \times (block\ size + tag\ size + valid\ field\ size) = 8 \times (64 + 28 + 1) = 744\ bits$$

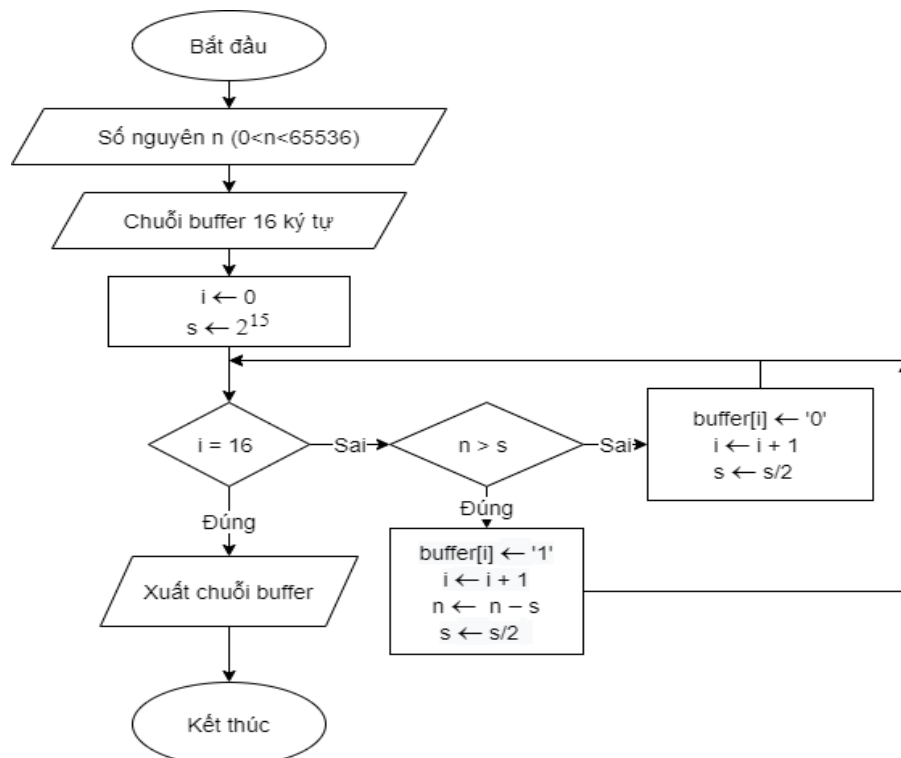
2 Giải pháp hiện thực

Giải pháp hiện thực câu 1:

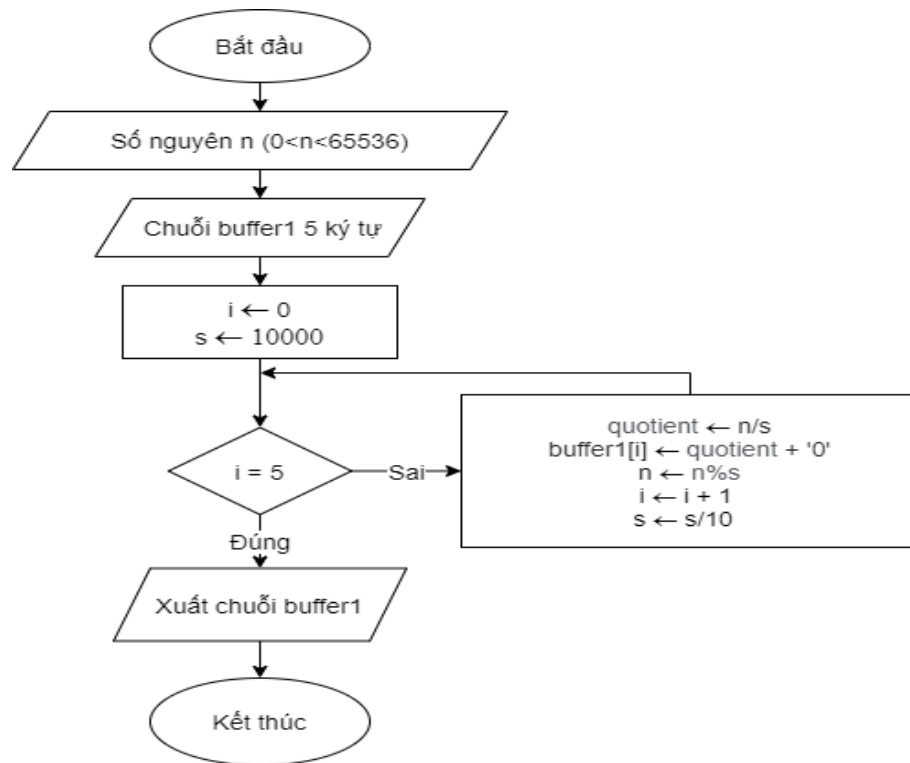
- B1: Sử dụng chức năng time (system time) (syscall 30) và chức năng set seed (syscall 40) để thiết lập ban đầu cho random.
- B2: Sử dụng chức năng random int range (syscall 42) để tạo ngẫu nhiên 1 số nguyên n trong khoảng 0 đến 65536.
- B3: Viết các hàm chuyển đổi số nguyên n thành chuỗi ký tự:
 - Hàm binary: chuyển đổi số nguyên n thành chuỗi ký tự số hệ 2 (16 ký tự).
 - Hàm decimal: chuyển đổi số nguyên n thành chuỗi ký tự số hệ 10 (5 ký tự).
 - Hàm hexadecimal: chuyển đổi số nguyên n thành chuỗi ký tự số hệ 16 (4 ký tự).
- B4: Sử dụng chức năng open file (syscall 13) mở file CHUOISO.TXT để lưu kết quả vào.
- B5: Viết hàm fputs và sử dụng chức năng write to file (syscall 15) để ghi kết quả vào file đã mở.
- B6: Sử dụng chức năng close file (syscall 16) đóng file.
- B7: Sử dụng chức năng print string (syscall 4) và chức năng print integer (syscall 1) để xuất kết quả ra màn hình.
- B8: Sử dụng chức năng exit (terminate execution) (syscall 10) kết thúc chương trình.

3 Giải thuật

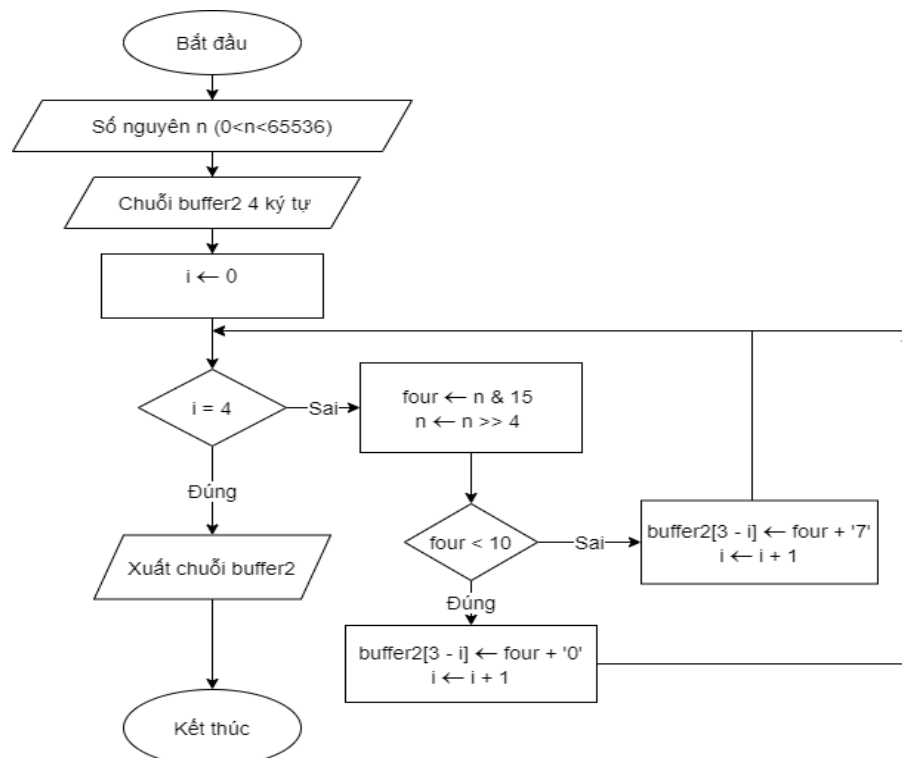
- Thuật toán đổi số nguyên n ($0 < n < 65536$) thành chuỗi ký tự số hệ 2 (16 ký tự):



- Thuật toán đổi số nguyên n ($0 < n < 65536$) thành chuỗi ký tự số hệ 10 (5 ký tự):

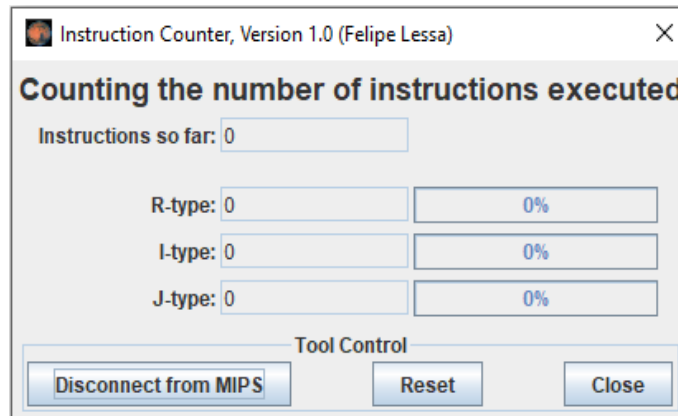


- Thuật toán đổi số nguyên n ($0 < n < 65536$) thành chuỗi ký tự số hệ 16 (4 ký tự):
 - Phép AND: $\&$
 - Phép dịch phải: \gg



4 Thống kê số lệnh, loại lệnh sử dụng trong chương trình

Ta sử dụng công cụ Instruction Counter trong MARS MIPS 4.5 để thống kê số lệnh trong chương trình:



- Tổng số lệnh thực thi (Instructions so far)
- Lệnh R (R-type)
- Lệnh I (I-type)
- Lệnh J (J-type)

5 Thời gian thực thi

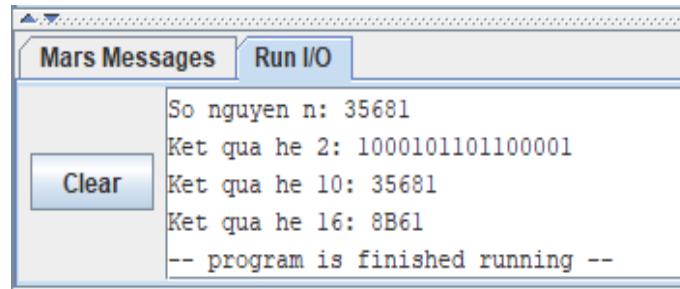
Để tính thời gian thực thi chương trình, ta sử dụng công thức:

$$CPUTime = \frac{CPUClockCycles}{ClockRate} = \frac{InstructionCount \times CPI}{ClockRate}$$

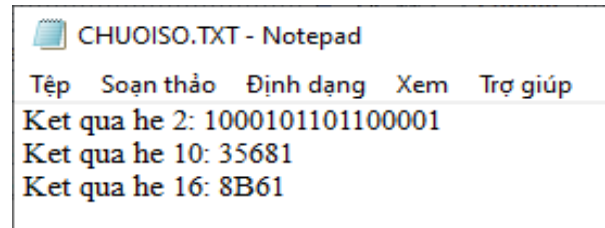
- CPU Time: Thời gian CPU xử lý chương trình.
- CPU Clock Cycles: Tổng chu kỳ thực thi.
- Clock Rate: Tần số (CR = 1GHz).
- Instruction Count: Tổng số lệnh thực thi của chương trình.
- CPI: Số chu kỳ thực hiện 1 lệnh (CPI = 1).

6 Kiểm tra chương trình

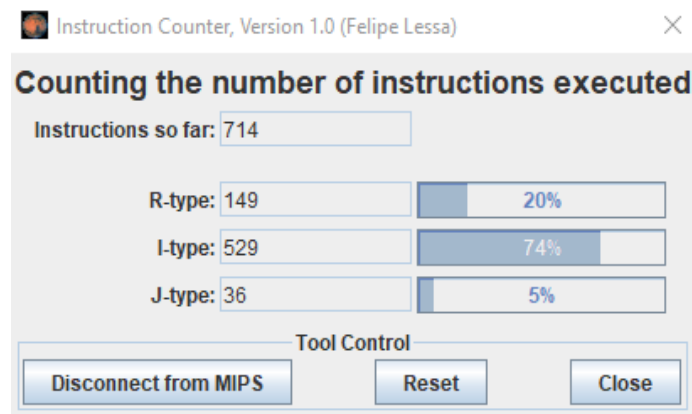
- Kiểm tra 1:
 - Kết quả xuất ra màn hình:



– Kết quả xuất ra tập tin CHUOISO.TXT:



– Số lệnh thực thi:

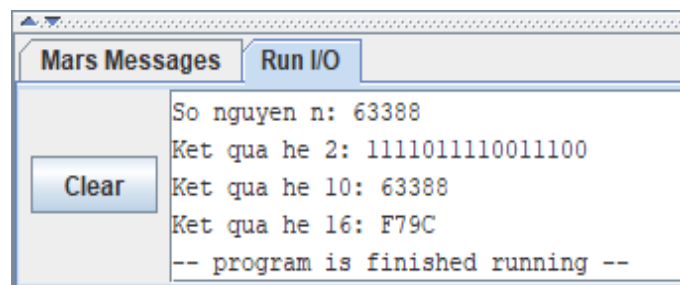


– Thời gian thực thi:

$$CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{714 \times 1}{10^9} = 7.14 \times 10^{-7} (s) = 714 (ns)$$

• Kiểm tra 2:

– Kết quả xuất ra màn hình:



– Kết quả xuất ra tập tin CHUOISO.TXT:

CHUOISO.TXT - Notepad

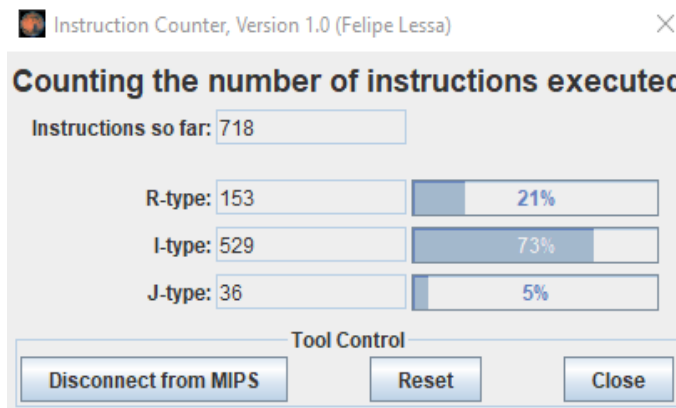
Tệp Soạn thảo Định dạng Xem Trợ giúp

Ket qua he 2: 1111011110011100

Ket qua he 10: 63388

Ket qua he 16: F79C

– Số lệnh thực thi:



– Thời gian thực thi:

$$CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{718 \times 1}{10^9} = 7.18 \times 10^{-7} (s) = 718 (ns)$$

7 Tổng kết

Qua bài tập lớn cá nhân này, em nắm vững hơn trong việc viết chương trình và sử dụng các chức năng trên MARS MIPS 4.5, hiểu rõ hơn về bộ nhớ cache cũng như việc chuyển đổi số nguyên thành các chuỗi ký tự số hệ 2, hệ 10 và hệ 16.

Tài liệu

- [1] David A. Patterson & John L. Hennessy – *Computer Organization and Design: The Hardware/ Software Interface*