



# COMPUTER ORGANISATION (TỔ CHỨC MÁY TÍNH)

---

## INTRODUCTION

# Acknowledgement

- The contents of these slides have origin from School of Computing, National University of Singapore.
- We greatly appreciate support from Mr. Aaron Tan Tuck Choy for kindly sharing these materials.

# Policies for students

- These contents are only used for students PERSONALLY.
- Students are NOT allowed to modify or deliver these contents to anywhere or anyone for any purpose.

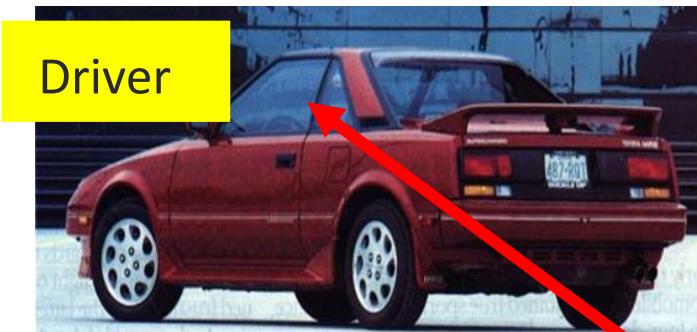
# Recording of modifications

- Minor change in slide 13 (replace new picture)

# Why Learn Computer Organisation? (From user to builder)

- You want to call yourself a computer scientist/specialist.
- You want to build software people use.
- You need to make purchasing decisions.
- You need to offer “expert” advice.
- Hardware and software affect performance
  - Algorithm determines number of source-level statements ([Programming methodology](#))
  - Language, compiler, and architecture determine machine instructions ([COD chapters 2 and 3](#))
  - Processor and memory determine how fast instructions are executed ([COD chapters 5, 6 and 7](#))
- Understanding performance ([COD chapter 4](#))

# So, what's a computer? (1/6)



Example: An automobile augments our power of locomotion.

A computer is a device capable of solving problems according to designed programs. It simply augments our power of storage and speed of calculation.



# So, what's a computer? (2/6)

- From **computer organisation** perspective, we study the **components** and **how they work together**
  - Processor, memory, input/output devices, networks, ...



Credit: <http://tech4abc.blogspot.sg/2010/08/latest-technology-in-computer-hardwares.html>

# So, what's a computer? (3/6)

MOST RECENT

July 8th, 2012

1 COMMENT »

## Most Recent Computer Technology

Written by: admin

Tags: breaking

f t +1 in



Do you know what is in your computer? Maybe you peeked when the repair technician was installing amazing for you. When you primary open up the CPU and seem inside, a computer is a very intimidating machine. But

once you are acquainted with about the dissimilar parts that make up a total computer it gets a lot easier. Today's computer consists of around eight main devices; some of the advanced computers might have a few additional mechanisms. What are these eight main components and what are they used for? We will start with beginner level facts to get you in progress.

First is the Power Supply. The authority provides is used to provide electrical

SHARE

Like 0 Digg

1. Power supply
2. Motherboard
3. **Central Processing Unit (CPU)**
4. Random Access Memory (RAM)
5. Hard drive
6. Cooling fan
7. I/O devices



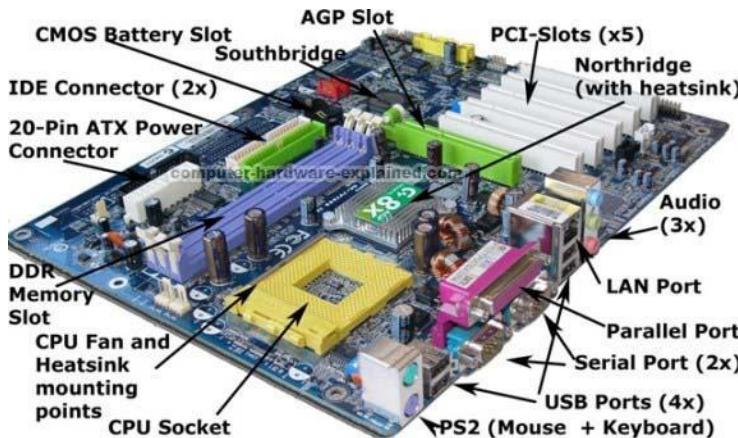
Credit:

[http://www.overclock3d.net/reviews/cpu\\_mainboard/the\\_computer\\_council\\_clocked\\_gamer\\_quad/1](http://www.overclock3d.net/reviews/cpu_mainboard/the_computer_council_clocked_gamer_quad/1)

Credit: <http://tech3news.com/most-recent-computer-technology/>

# So, what's a computer? (4/6)

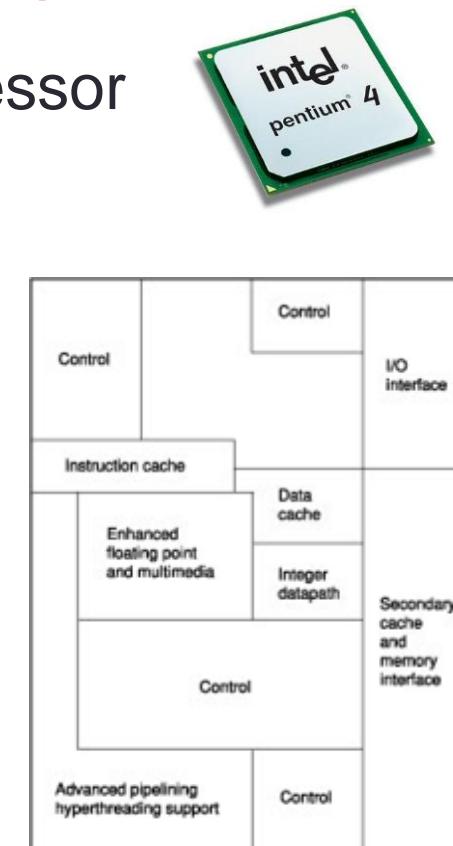
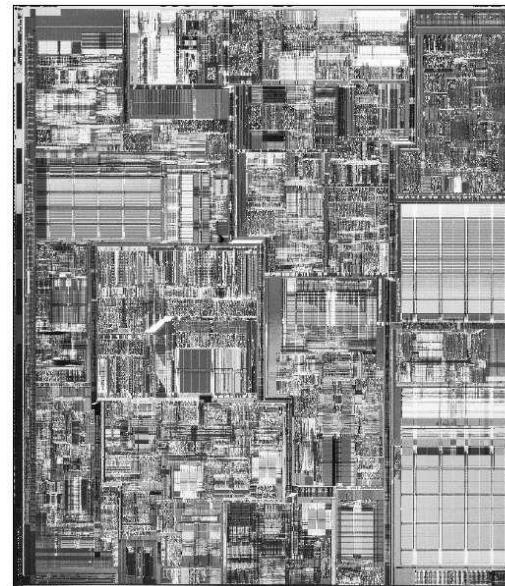
- PC motherboard



Credit: <http://www.computer-hardware-explained.com/what-is-a-motherboard.html>

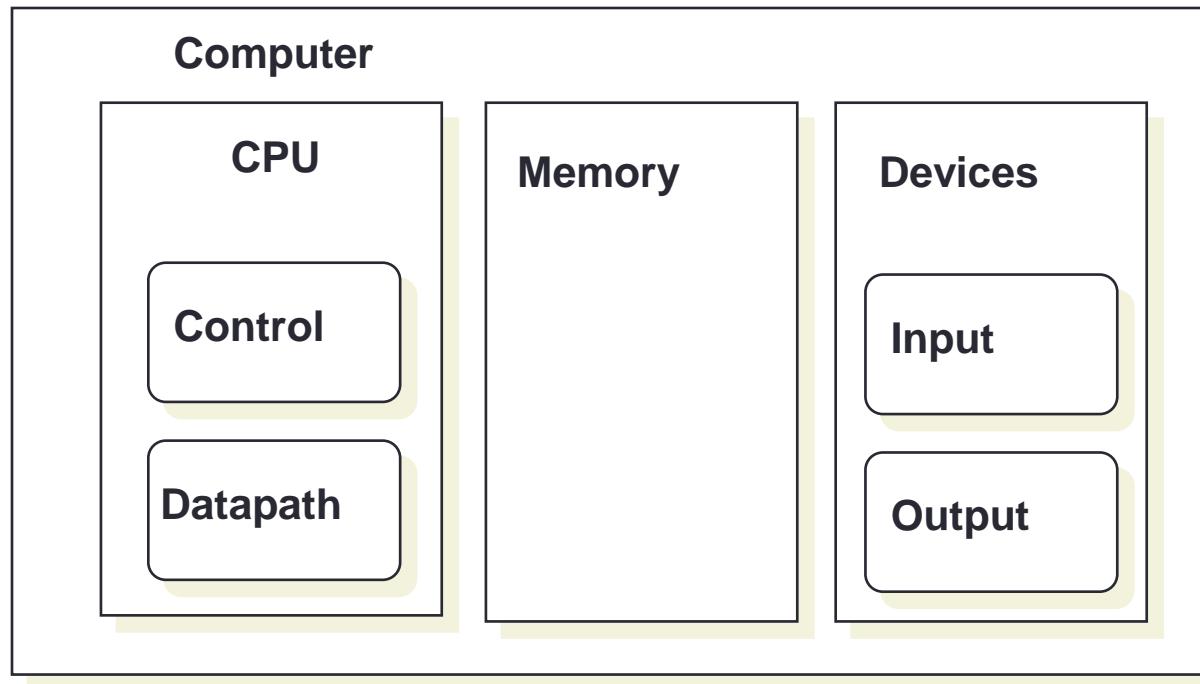
- Pentium processor

Inside a Pentium chip



# So, what's a computer? (5/6)

## ■ Computer Organisation



# So, what's a computer? (6/6)

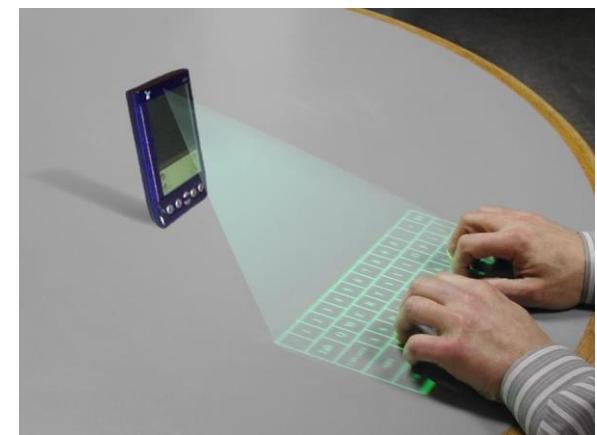
- Next generation...



Credit:  
<http://www.prabhanjamindiaits.com/blogdetailedpage.aspx?id=66>



Credit: <http://www.custom-build-computers.com/Latest-Computer-Hardware.html>



Credit: <http://new-techpc.blogspot.sg/2012/10/latest-in-computer-technology.html>

# 6<sup>th</sup> January 2014

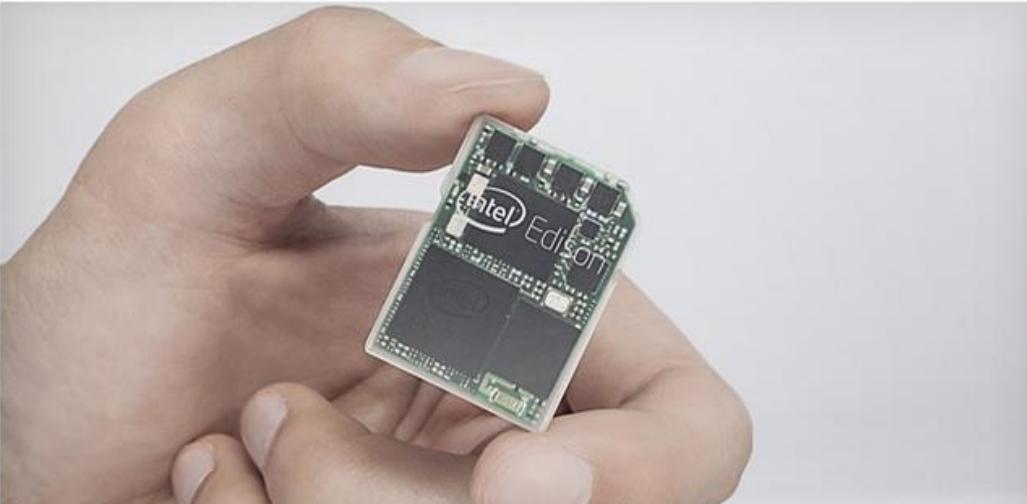
[www.theverge.com/2014/1/6/5282472/intel-announces-edison-a-computer-the-size-of-an-sd-card](http://www.theverge.com/2014/1/6/5282472/intel-announces-edison-a-computer-the-size-of-an-sd-card)



## Intel announces Edison, a computer the size of an SD card

By Sean Hollister on January 6, 2014 10:01 pm Email

DON'T MISS STORIES [FOLLOW THE VERGE](#) 8+ Like 280k Follow 355K followers



<http://www.theverge.com/2014/1/6/5282472/intel-announces-edison-a-computer-the-size-of-an-sd-card>

PART OF THIS STORYSTREAM

# ABSTRACTION (1/2)

- Delving into depth reveals more information
- Abstraction omits unnecessary details

High-level Language

Assembly Language

Machine Language

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
TEMP = V(K)
V(K) = V(K+1)
V(K+1) = TEMP
```

C/Java Compiler

Fortran Compiler

```
lw $t0, 0($2)
lw $t1, 4($2)
sw $t1, 0($2)
sw $t0, 4($2)
```

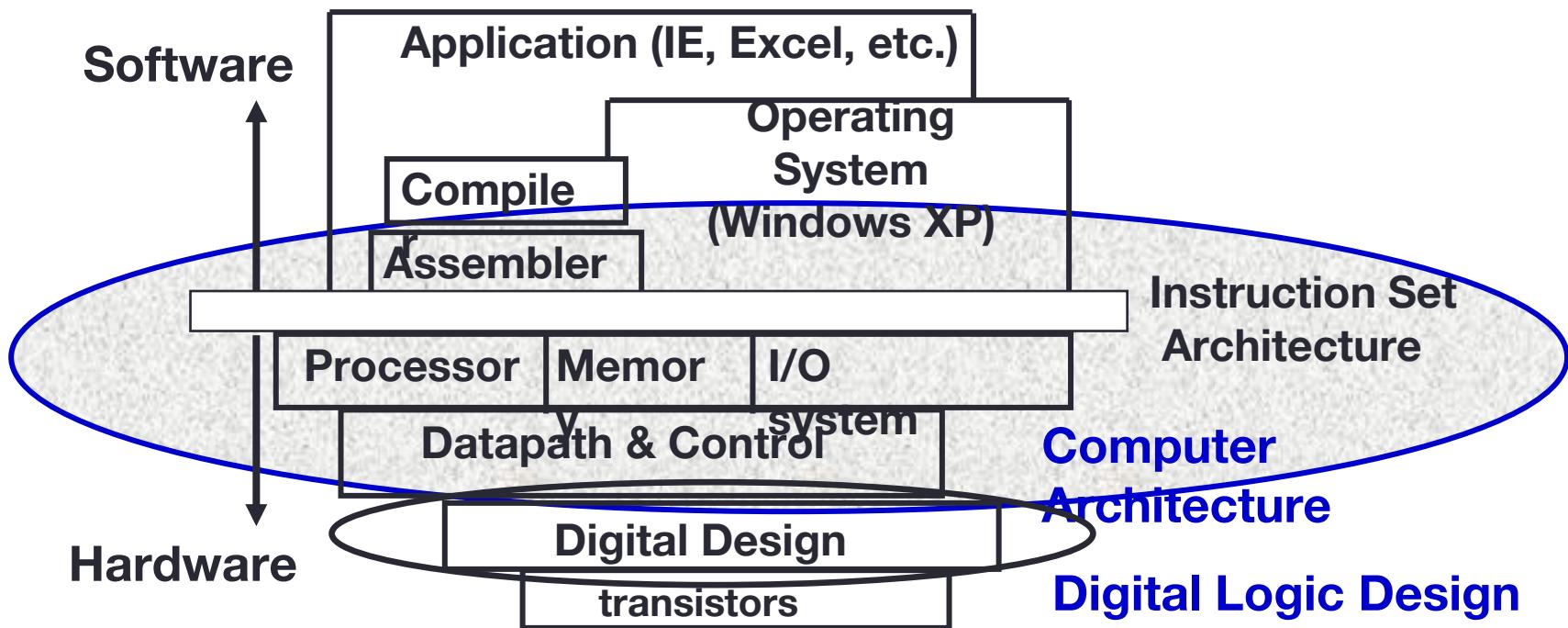
MIPS Assembler

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

# ABSTRACTION (2/2)

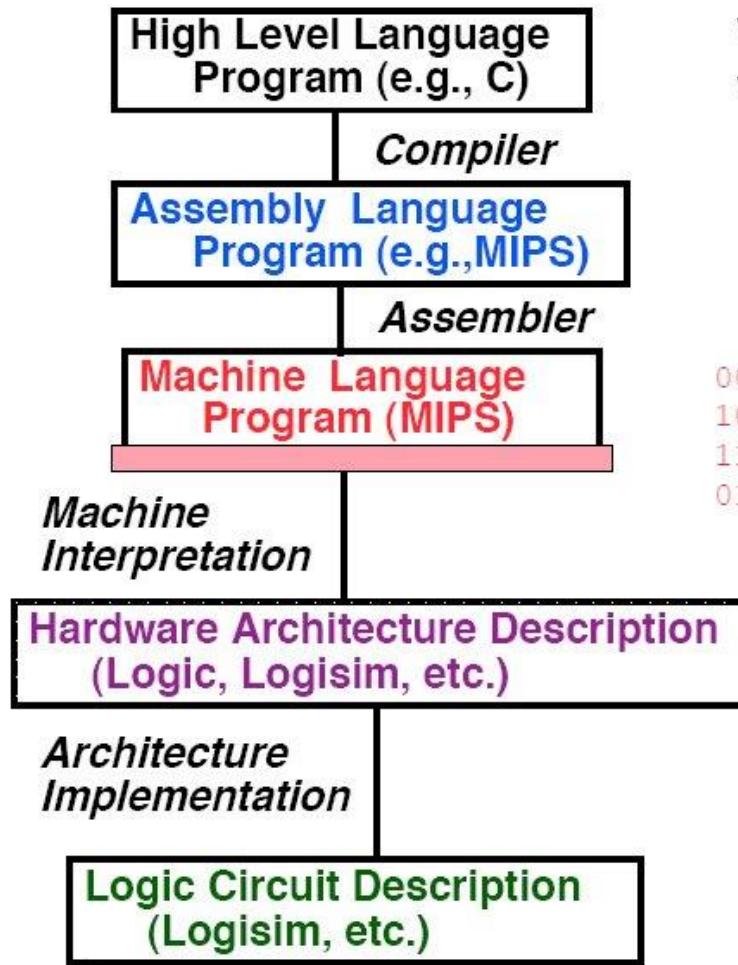
- Need to learn abstractions such as:
  - Application and system software
  - Assembly language and machine language
  - Architectural issues such as pipelining, caches, virtual memory
  - Combinational logic, arithmetic circuits
  - Sequential logic, finite state machines
  - Boolean logic (1s and 0s)
  - Transistors used to build logic gates (CMOS - Complementary Metal Oxide Semiconductors)
  - Semi-conductors/silicon used to build transistors
  - Properties of atoms, electrons and quantum dynamics
- So much to learn!

# HOW DO THE PIECES FIT TOGETHER?



- Coordination of many *levels of abstraction*
- Under a rapidly *changing set of forces*
- Design, measurement, and evaluation

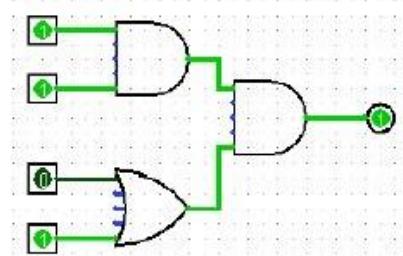
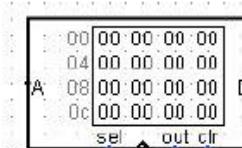
# LEVELS OF REPRESENTATION



`temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;`

`Iw $t0, 0($2)  
Iw $t1, 4($2)  
Sw $t1, 0($2)  
Sw $t0, 4($2)`

0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111

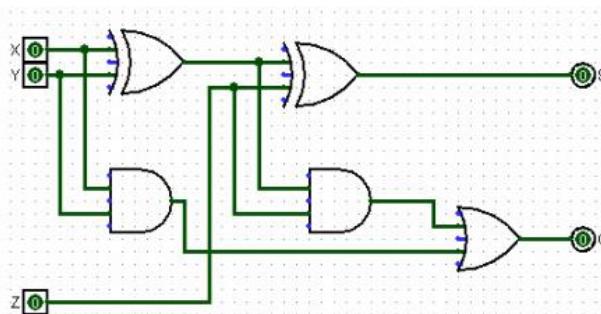


# SYLLABUS OUTLINE (REVISIT)

- Number systems and codes
  - Boolean algebra
  - Logic gates and circuits
  - Simplification
  - Combinational circuits
  - Sequential circuits
  - Performance
  - Assembly language
  - The processor: Datapath and control
  - Pipelining
  - Memory hierarchy: Cache
  - Input/output
- 
- The diagram illustrates the syllabus outline with three main categories, each enclosed in a blue curly brace:
- Preparation: 2 weeks** (Includes: Number systems and codes, Boolean algebra, Logic gates and circuits, Simplification, Combinational circuits, Sequential circuits)
  - Logic Design: 3 weeks** (Includes: Performance, Assembly language, The processor: Datapath and control, Pipelining, Memory hierarchy: Cache, Input/output)
  - Computer organisation** (Includes: All topics under Preparation and Logic Design)

# PRACTICAL ASPECTS

- Practical experience
  - Logic design experiments
  - Logisim software
  - PC Spim software



PCSpim

File Simulator Window Help

PC	- 00000000	EPC	- 00000000	Cause	- 00000000	BadAddr= 00000000
Status	- 3000ff10	HI	- 00000000	LO	- 00000000	
General Registers						
R0 (r0)	- 00000000	R8 (t0)	- 00000000	R16 (s0)	- 00000000	R24 (t8) = 00000000
R1 (at)	- 00000000	R9 (t1)	- 00000000	R17 (s1)	- 00000000	R25 (t9) = 00000000
R2 (v0)	- 00000000	R10 (t2)	- 00000000	R18 (s2)	- 00000000	R26 (k0) = 00000000
R3 (v1)	- 00000000	R11 (t3)	- 00000000	R19 (s3)	- 00000000	R27 (k1) = 00000000
R4 (a0)	- 00000000	R12 (t4)	- 00000000	R20 (s4)	- 00000000	R28 (gp) = 10000000

```

PC          - 00000000    EPC      - 00000000    Cause   - 00000000    BadAddr= 00000000
Status     - 3000ff10    HI       - 00000000    LO      - 00000000
General Registers
R0 (r0) - 00000000    R8 (t0) - 00000000    R16 (s0) - 00000000    R24 (t8) = 00000000
R1 (at) - 00000000    R9 (t1) - 00000000    R17 (s1) - 00000000    R25 (t9) = 00000000
R2 (v0) - 00000000    R10 (t2) - 00000000   R18 (s2) - 00000000    R26 (k0) = 00000000
R3 (v1) - 00000000    R11 (t3) - 00000000   R19 (s3) - 00000000    R27 (k1) = 00000000
R4 (a0) - 00000000    R12 (t4) - 00000000   R20 (s4) - 00000000    R28 (gp) = 10000000
[0x00400000] 0x8fe40000 lw $4, 0($29) ; 174: lw $e0 0($sp)
[0x00400004] 0x27e50004 addiu $5, $29, 4 ; 175: addiu $a1 $sp 4
[0x00400008] 0x24e60004 addiu $6, $5, 4 ; 176: addiu $a2 $a1 4
[0x0040000c] 0x00041080 sll $2, $4, 2 ; 177: sll $v1 $a0 2
[0x00400010] 0x00c23021 addi $6, $6, $2 ; 178: addu $a2 $a2 $v0
[0x00400014] 0x0c000000 jal 0x00000000 [main] ; 179: jal main
[0x00400018] 0x00000000 nop ; 180: nop
[0x0040001c] 0x3402000a ori $2, $0, 10 ; 182: li $v0 10

```

DATA  
[0x10000000]...[0x10040000] 0x00000000

STACK  
[0x7ffffeffc] 0x00000000

KERNEL DATA  
[0x90000000] 0x78452020 0x74706563 0x206e6f69 0x636f2000

SPIN Version Version 7.0 of July 7, 2004  
Copyright 1990-2004 by James R. Larus (larus@cs.wisc.edu).  
All Rights Reserved.  
DOS and Windows ports by David A. Carley (dac@cs.wisc.edu).  
Copyright 1997 by Morgan Kaufmann Publishers, Inc.  
See the file README for a full copyright notice.  
Loaded: C:\Program Files\PCSpim7\exceptions.s

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

# QUOTES

- “What I hear, I forget.  
What I see, I remember.  
What I do, I understand”
- “The important thing is to understand what you are doing,  
rather than to get the right answer” – Tom Lehrer

# Q&A