

程序代写代做 CS 编程辅导

# COMP2300/6300

Computer Organisation and Programming



Architecture and Organisation

Dr Charles Martin

Semester 1, 2022



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Admin Time

程序代写代做 CS编程辅导

Assignment 2 due *Sunday 22/5/2022*



AEST

Lab 11 starts tomorrow – keep going to labs!

WeChat: estutorcs

Lab Pack 4 due: Wednesday 1/6/2022 23:59 AEST (day before exam period)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



## Week 11: Architecture

WeChat: cstutorcs

Assignment Project Exam Help

What is a computer again?

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Outline

程序代写代做 CS编程辅导

- a basic computer
- origins
- architectures
- alternatives?
- visual ARM1



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A basic computer...

程序代写代做 CS编程辅导



## R6502 FEATURES

- 64K addressable bytes of memory (A0–A15)
- On-chip clock  
TTL-level single phase input  
RC time base input  
crystal time base input
- Two phase output clock for timing of support chips
- $\overline{\text{IRQ}}$  interrupt
- $\overline{\text{NMI}}$  interrupt
- $\overline{\text{RDY}}$  signal
- SYNC signal  
(can be used for single instruction execution)
- 40-pin DIP

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Don't we need more stuff?

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

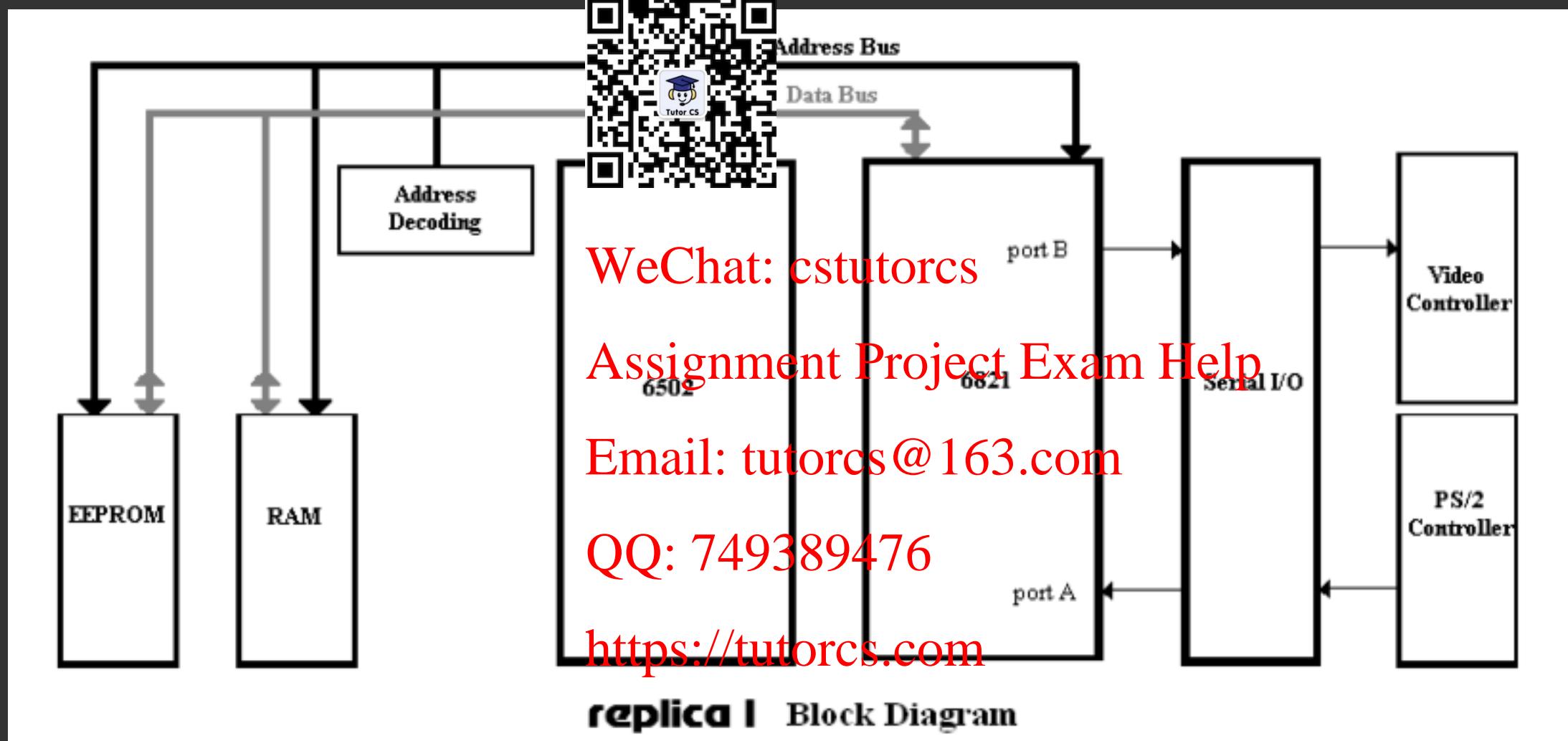
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Don't we need more stuff? 程序代写代做 CS编程辅导



程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Origins

---

# The Z1 (1937)

created by Konrad Zuse

程序代写代做 CS编程辅导



first digital computer: relays, programmable via punch tape, 1 Hz clock, 64 words of memory @ 22-bit, 2 registers, floating point unit, weight 1 ton

image from ComputerGeek, CC BY-SA 3.0

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Z3 (1941)

first freely programmable (Turin)



程序代写代做 CS 编程辅导

继电器计算机，5.3 Hz 时钟

image from Venusianer, CC BY-SA 3.0

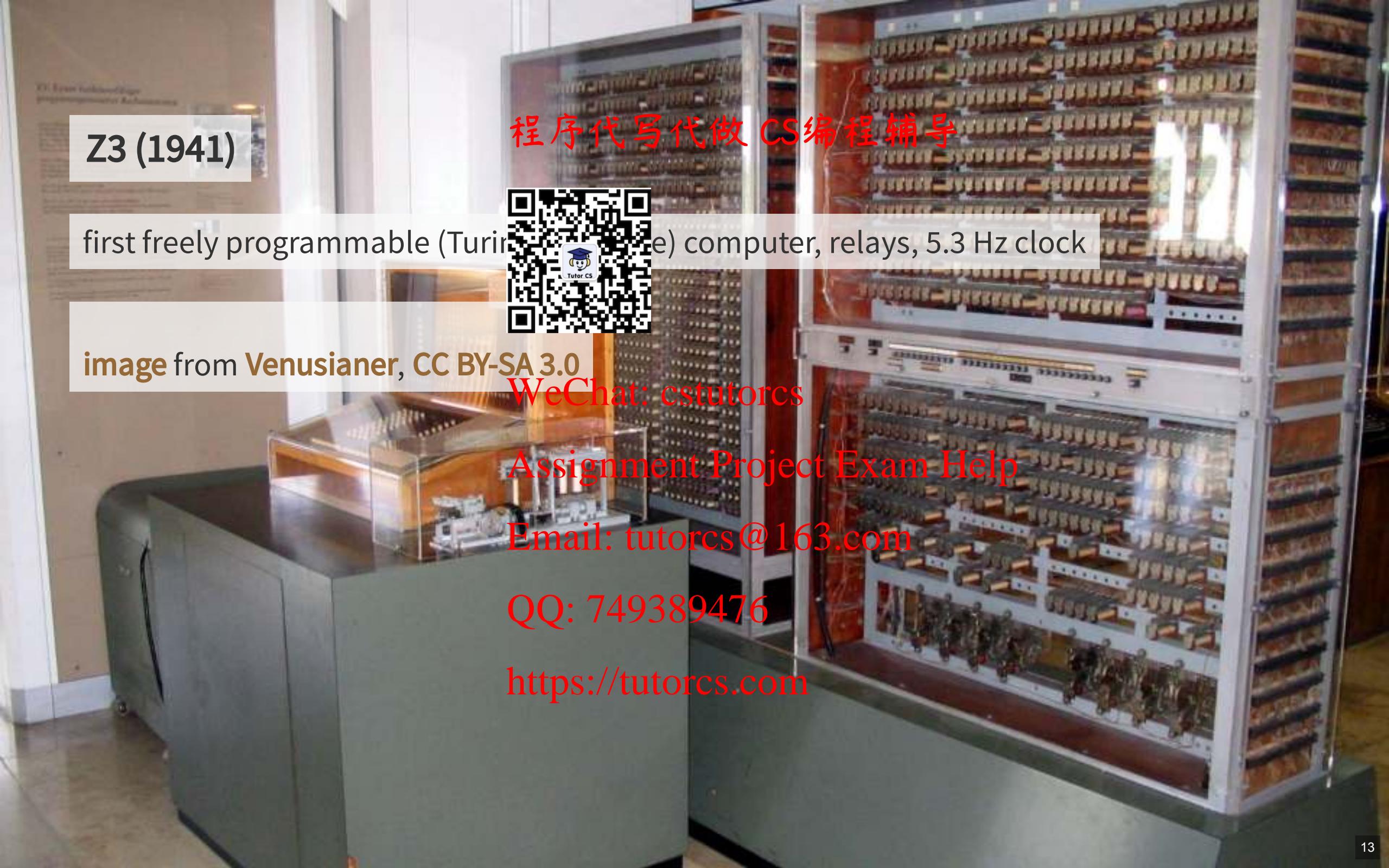
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# ENIAC (1945)

程序代写代做 CS编程辅导

ENIAC: first Turing complete vacuum tube computer, 100 kHz clock, weight 27 tons, size 167 sq m



computer, 100 kHz clock, weight 27 tons, size

image from U.S. Army Photo, public domain

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



... then came the microbit

Assignment Project Exam Help  
*(not really)*

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Architectures

---

## Harvard architecture

程序代写代做 CS 编程辅导

control unit concurrently addresses program and data memory and fetches next instruction  
—controls next ALU operations and updates condition based on ALU status



Arithmetic Logic Unit (ALU) fetches data from memory,  
executes arithmetic/logic operations, and writes data  
to memory

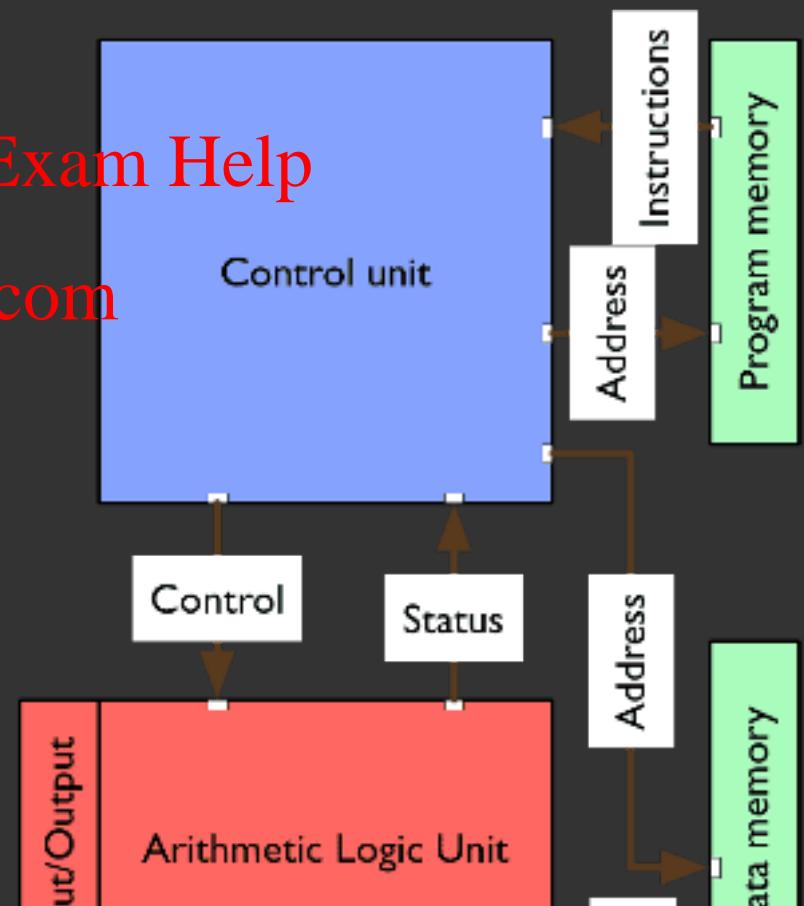
separate memory for program & data

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# von Neumann architecture 程序代写代做 CS 编程辅导

control unit sequentially addresses program and data memory and fetches next instruction  
—controls next ALU operations and provides condition based on ALU status



**Arithmetic Logic Unit (ALU)** fetches data from memory,  
executes arithmetic/logic operations, and writes data  
to memory

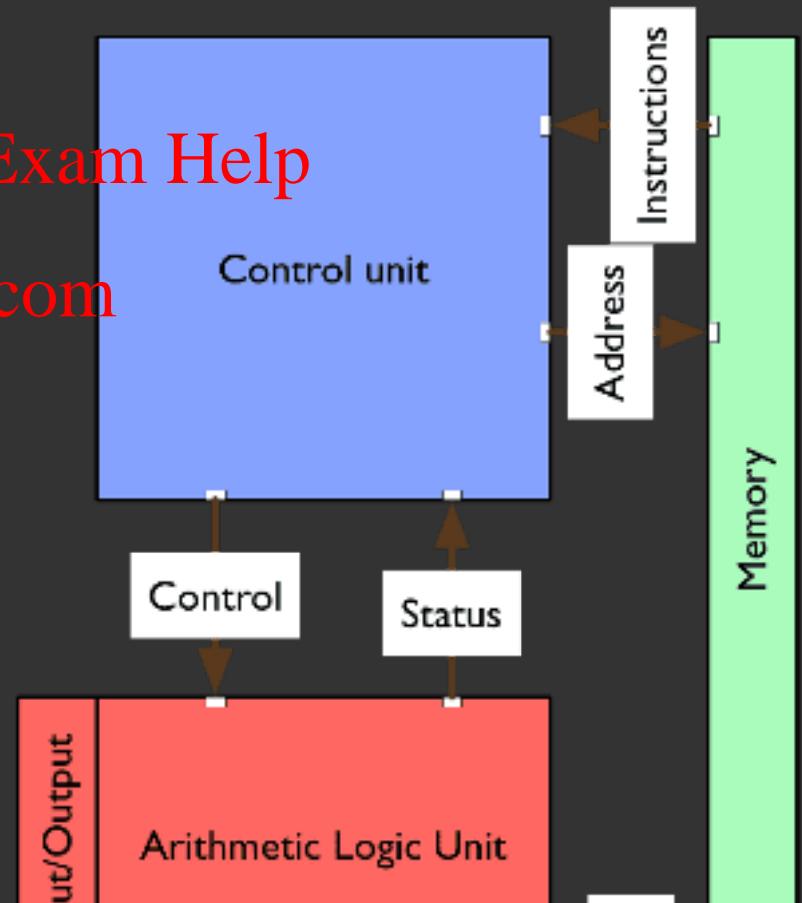
program and data memory not distinguished (so  
programs can change themselves)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# A simple CPU

程序代写代做 CS编程辅导

decoder/sequencer converts instructions to CPU control signals



arithmetic logic unit (ALU) performs arithmetic & logic operations

registers provide small, fast storage to the CPU

WeChat: cstutorcs

flags indicate the states of the latest calculations

Assignment Project Exam Help

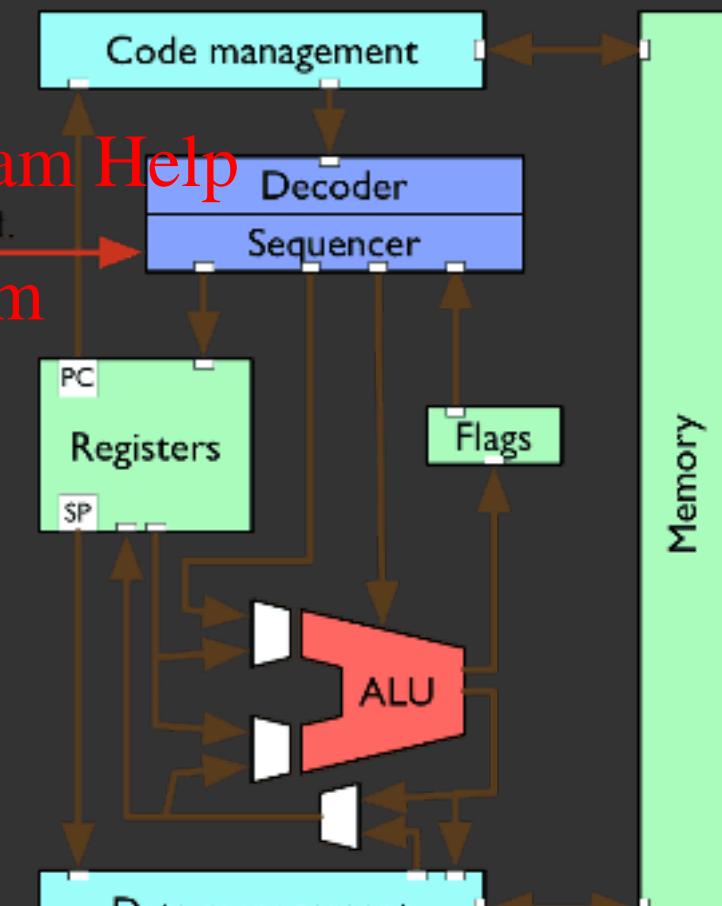
Email: tutorcs@163.com

code/data management for loading/storing, caching

memory

QQ: 749389476

<https://tutorcs.com>



# Pipeline

程序代写代做 CS编程辅导

some CPU actions are naturally sequential (e.g. **fetch-decode-execute**).



an **instruction pipeline** allows these sequential processes to be overlapped in *time*

same latency, but higher throughput

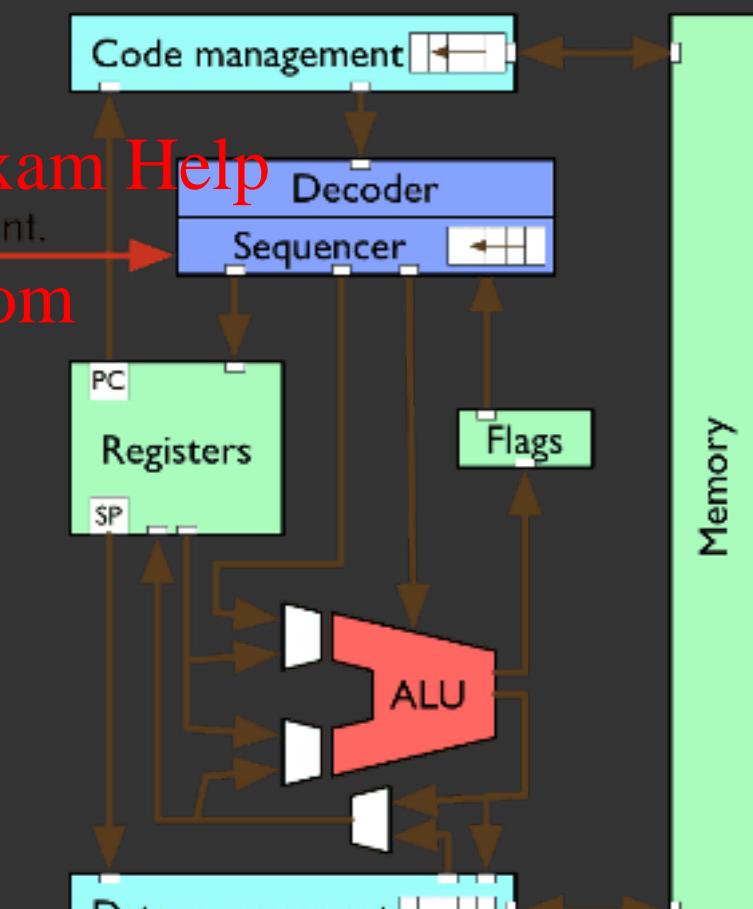
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Simple Pipeline

程序代写代做 CS编程辅导



Clock Cycles  
WeChat: cstutorcs  
Assignment Project Exam Help

Instructions	1	2	3	4	5
1	Fetch	Decode	Execute		
2		Fetch	Decode	Execute	
3			Fetch	Decode	Execute

Email: tutorcs@163.com

QQ: 749389476  
Fetch  
<https://tutorcs.com>

# Pipeline example

程序代写代做 CS编程辅导



# Pipeline Hazards

程序代写代做 CS编程辅导

- **Hazards** are circumstances that prevent the pipeline from working efficiently.
- **Data:** When the outcome of one instruction is required to execute the next.
- **Structural:** When part of the CPU hardware required by two instructions simultaneously (e.g., memory to fetch and store simultaneously).
- **Control:** When the location of the next branch is unknown until an instruction is executed.



WeChat: cstutorcs

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# How do you deal with these? 程序代写代做 CS编程辅导

- Add “bubbles” in the pipeline?
- branch prediction?
- out-of-order execution?
- all these have **downsides**



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# NO YOU'RE OUT OF ORDER

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## THIS WHOLE DAMN CPU'S OUT OF ORDER

## Out-of-order execution

程序代写代做 CS编程辅导

re-ordering the sequence inside



one leads to ‘out of order’ CPU designs

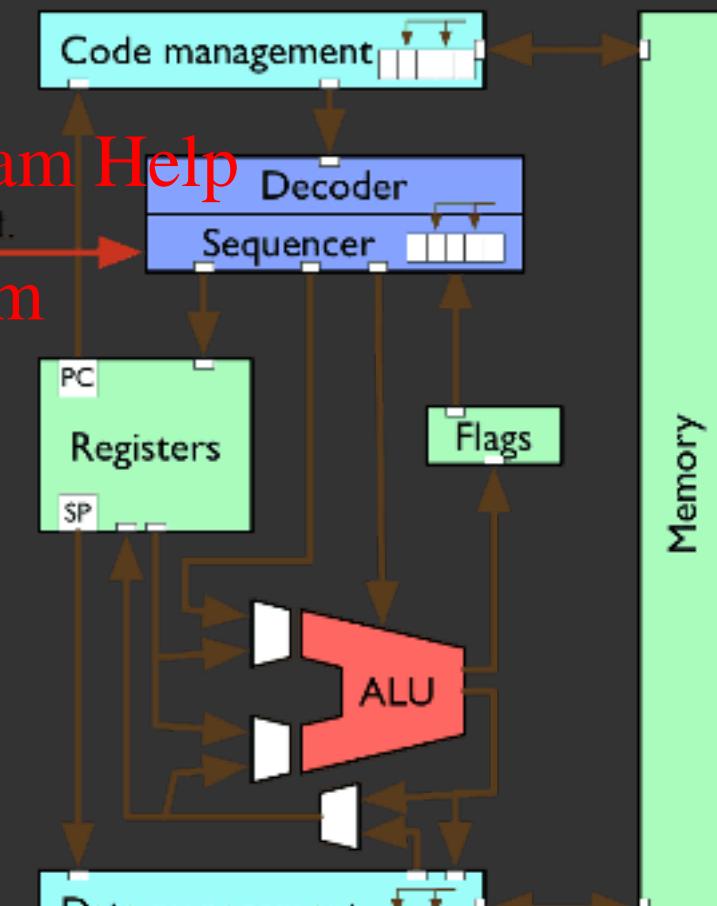
pipeline becomes a hardware scheduler, and results  
need to be “re-sequentialised” (or possibly discarded)

finer-grained sequences can be introduced by breaking  
CPU instructions into micro code (better if there are  
more independent instructions)

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



## Out-of-order example

$$f(a, b, c) = (a + b) \times c$$

```
@ in-order
ldr r0, =0x20000000
ldr r1, [r0] @ load a into r1
ldr r2, [r0, 4] @ load b into r2
ldr r3, [r0, 8] @ load c into r3
add r1, r1, r2 @ add a and b
mul r0, r1, r3 @ (a + b) * c
```



程序代写代做 CS编程辅导

```
@ out-of-order
ldr r0, =0x20000000
ldr r1, [r0]
ldr r2, [r0, 4]
add r1, r1, r2 @ these two are
ldr r3, [r0, 8] @ switched around
mul r0, r1, r3
```

Computerphile has a more detailed explanation

Email: tutorcs@163.com  
QQ: 749389476

<https://tutorcs.com>

it's not *quite* magic, but...

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

## Multiprocessing

---

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## SIMD/vector processing

程序代写代做 CS编程辅导

Single Instruction Multiple Data / SIMD  
multiple “datas” concurrently



often requires special “wide” registers and new  
instructions (e.g. fitting 4 32-bit values into one 128-bit  
register, then adding them all using one SIMD  
instruction)

requires specialised compilers or programming  
languages with implicit concurrency

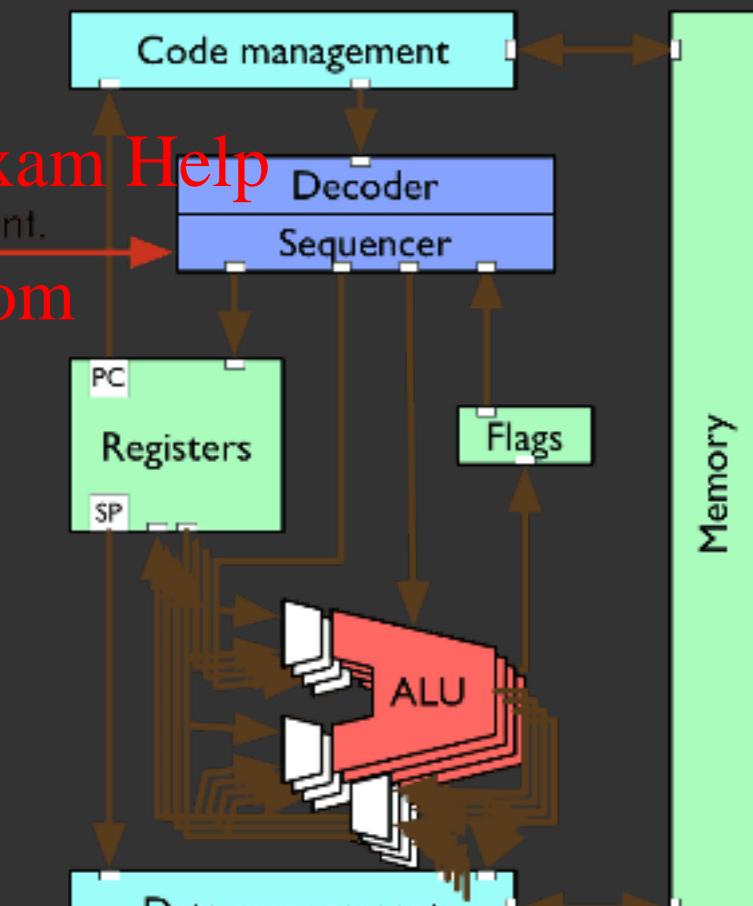
WeChat: cstutorcs  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

examples: NEON, Altivec, MMX, SSE/3DNow!, AVX

<https://tutorcs.com>



# Hyper-threading

程序代写代做 CS编程辅导

emulates multiple “virtual” CPU



- register sets
- decoder/sequencer
- flags
- interrupt logic

while sharing other resources like the ALU, data  
management

WeChat: cstutorcs

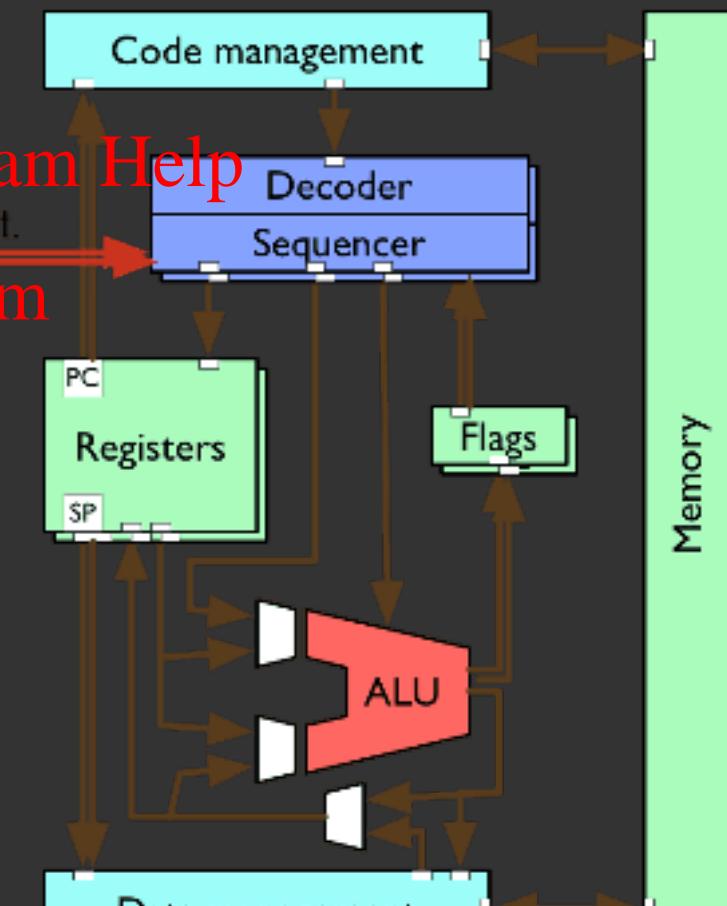
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

examples: Intel Core i-series, POWER9 (up to 8 threads per core)



talk

## 程序代写代做 CS编程辅导

what sort of workloads would be



SIMD? how about hyperthreading?

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Multi-core CPUs

程序代写代做 CS编程辅导

full replication of multiple CPU cores in the same chip package

often combined with the other techniques discussed

cleanest and most explicit implementation of concurrency on the CPU level

WeChat: cstutorcs

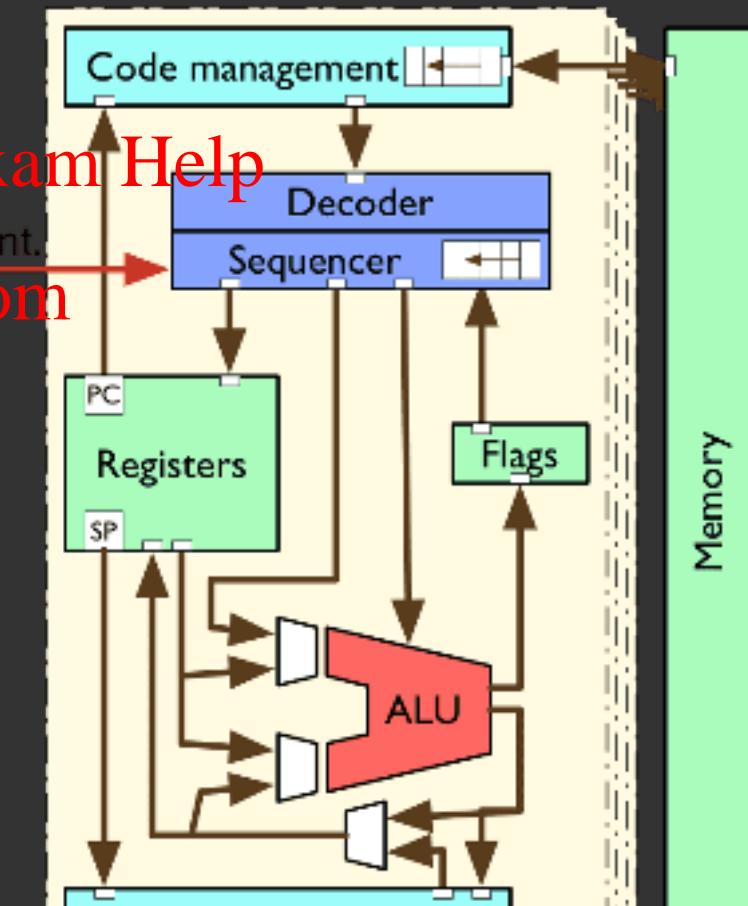
requires synchronised atomic operations, and programming languages with implicit or explicit concurrency

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# Flynn's Taxonomy (1966) 程序代写代做 CS编程辅导

Classification system for processes based on two characteristics:

single instruction



multiple data

single instruction

SISD - uniprocessing

SIMD (SSE, NEON)

multiple instruction

MISD? (??)  
**WeChat: cstutorcs**

MIMD (multi-core processors)

Which of these are common today?

**Assignment Project Exam Help**

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

**Amazing parallel computers of today**  
Assignment Project Exam Help

---

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Your phone!

程序代写代做 CS 编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# ARM big.LITTLE architecture



Heterogeneous computing archi

“little” cores: efficient and power-efficient (use these most of the time)

“big” cores: powerful but power-hungry (use these just in bursts)

WeChat: cstutorcs

all cores have access to same memory regions

Assignment Project Exam Help

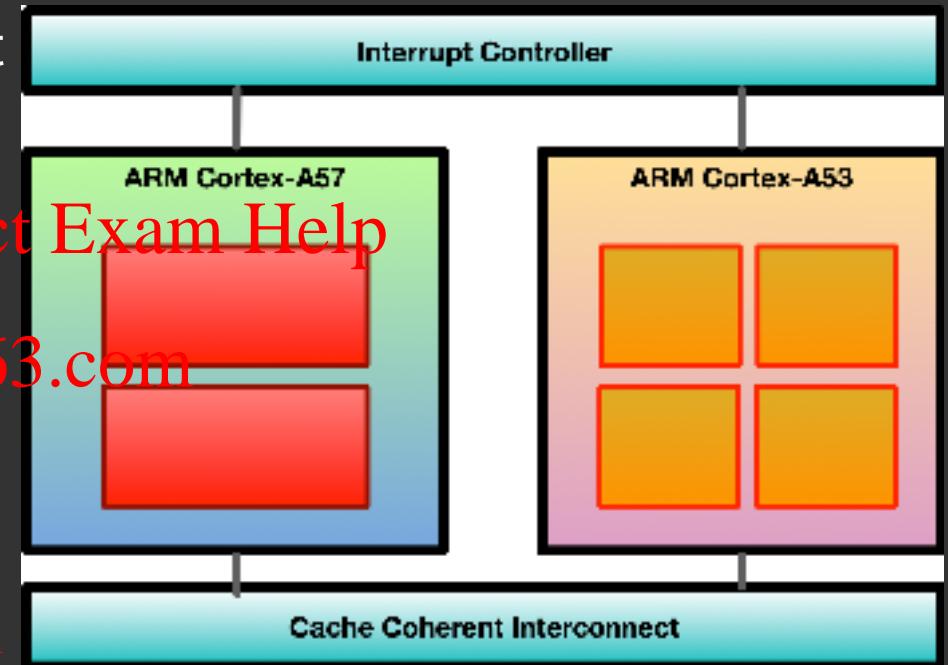
scheduler figures out where to execute threads

Email: tutorcs@163.com

QQ: 749389476

e.g., Apple A11 (2 big cores, 4 little)

<https://tutorcs.com>



Your gaming PC!

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# NVIDIA GPU architectures 程序代写代做 CS编程辅导

GPUs are **highly parallel** process



focussed on bandwidth, not latency

NVIDIA uses **CUDA (computer unified device architecture)**.

WeChat: cstutorcs

One CUDA SM (streaming multiprocessor) has **lots of cores**  
(~ALUs)

Assignment Project Exam Help

Image: **Copyright NVIDIA.**

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



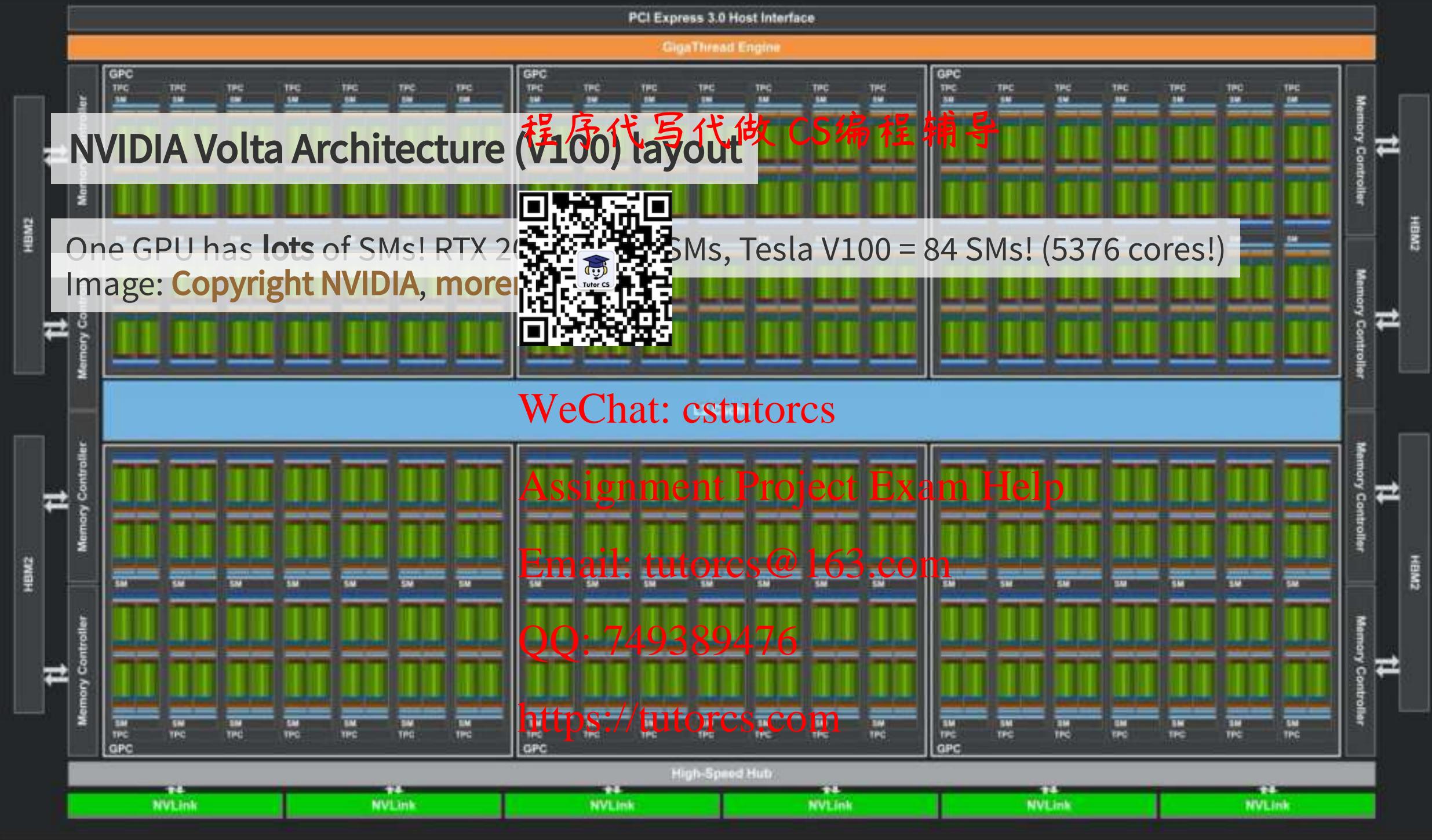


Figure 4. Volta GV100 Full GPU with 84 SM Units

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Alternatives?

---

Digital Computing in Plastic!

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

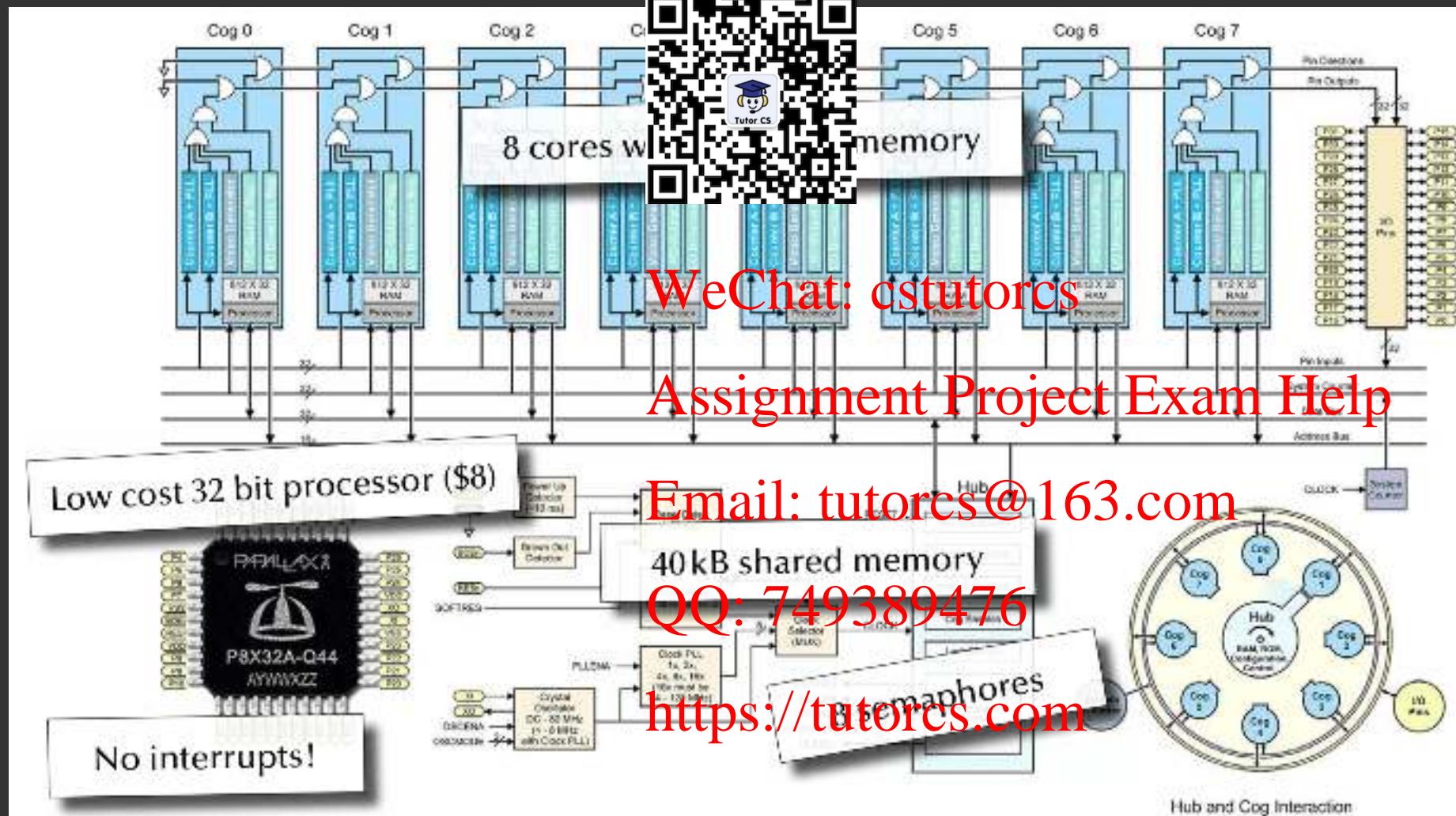
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

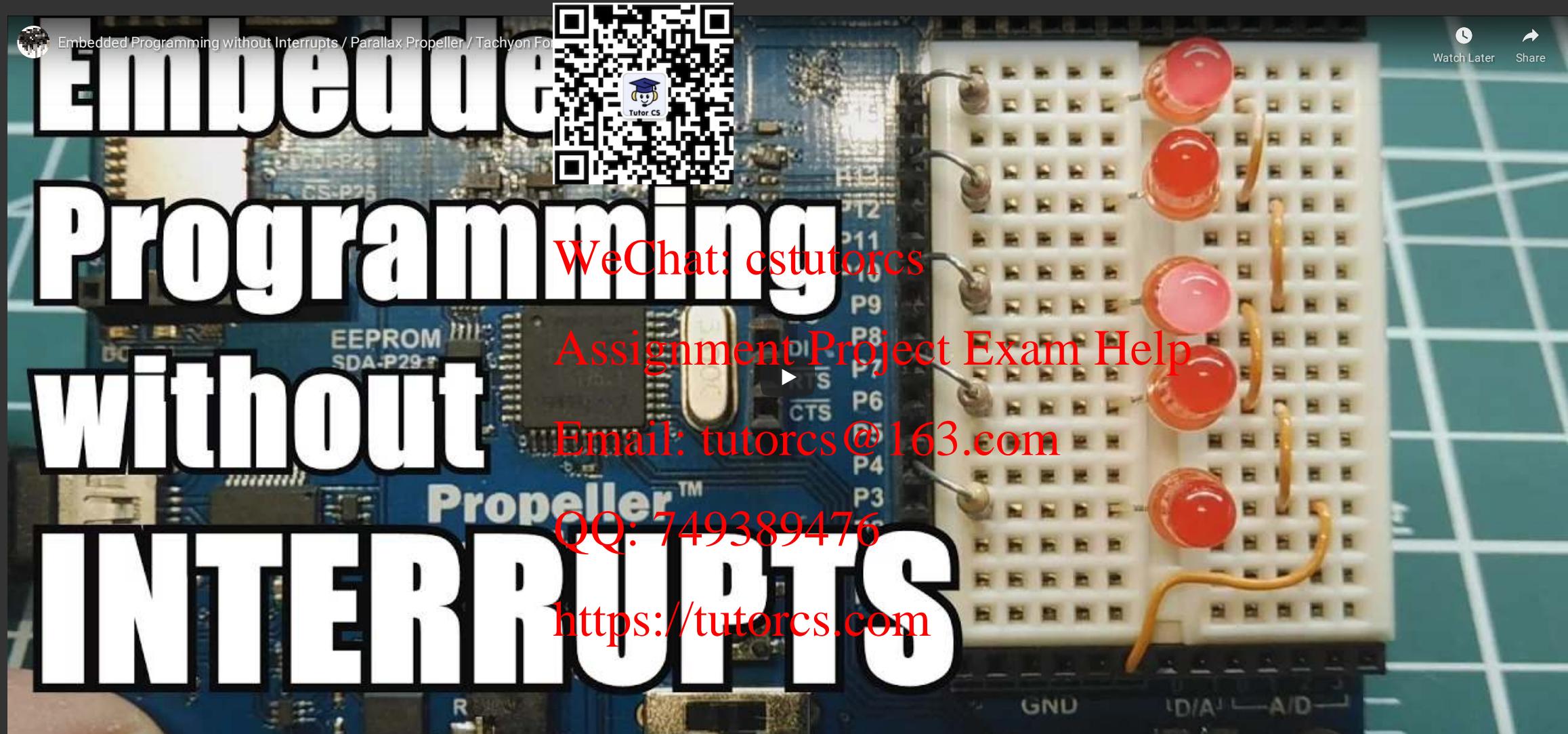
# Parallax Propeller

程序代写代做 CS编程辅导



# Embedded Programming without Interrupts

程序代写代做CS编程辅导



RISC-V

程序代写代做 CS 编程辅导

<https://riscv.org/>



*The Free and Open RISC Instruction Set Architecture*

a collaborative design effort

small core ISA with lots of optional “extensions” (e.g.  
floating point, SIMD/vector ops)

this allows it to scale from embedded to HPC (in  
*principle*)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Intel x86/amd64?

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## Visual ARM1 simulation

程序代写代做 CS编程辅导

ARM1 chip (created in 1985) is the



ancestor of all smartphone chips, and also the microbit

check out this **visual ARM1 simulation** (runs in the web browser)

WeChat: cstutorcs

also see **Ken Shirriff's explanation**

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

do you (**kindof**) get it?  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Replica 1 Demo!

replica I  
plus

www.brielcomputers.com



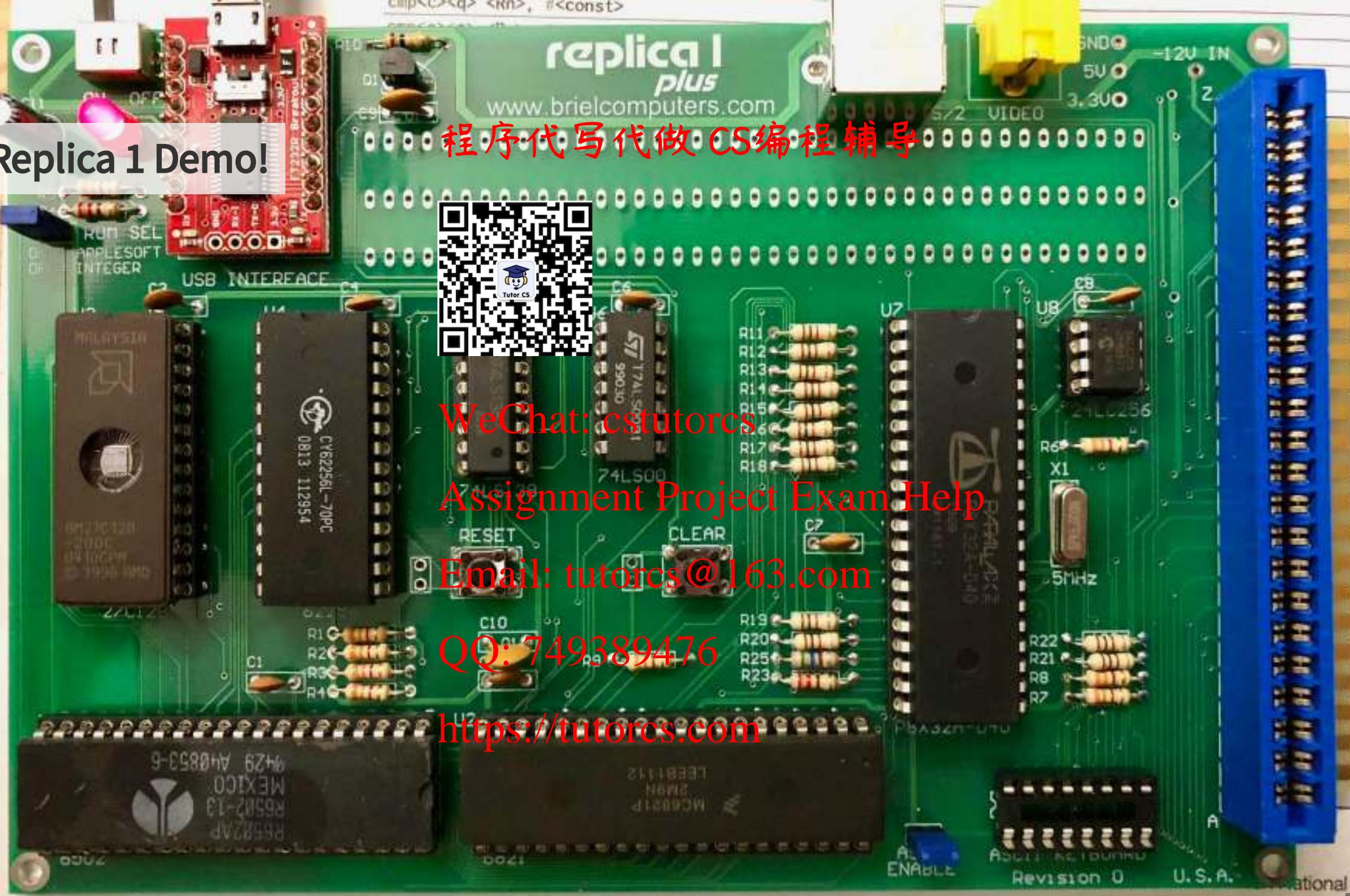
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



## Replica 1 Demo

程序代写代做 CS编程辅导

Let's have a look at an Apple I... (975)



- modern replica of Apple I computer

- **Serial Programming**

- No OS (has a “monitor” program and BASIC interpreter in ROM).

- MOS6502 processor, **6821 peripheral interface adapter** (PIA), one bank of RAM, one ROM

- Modern things: USB serial adaptor, Parallax Propeller (drives video and manages keyboard).

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476

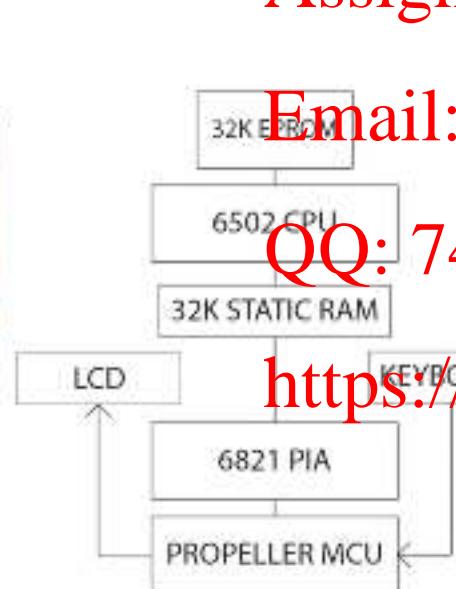
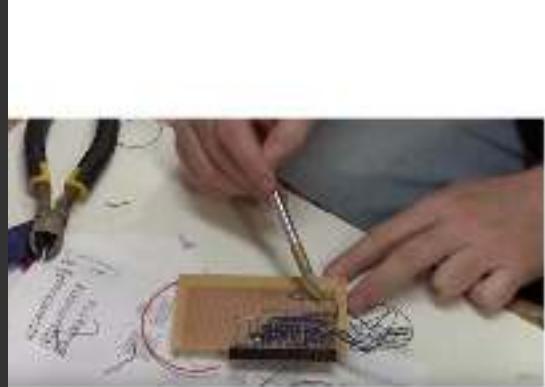
<https://tutorcs.com>

Make your own?

程序代写代做 CS编程辅导

- Ben Heck Apple 1 Replica Build
- Get parts from AliExpress/Ebay, soldering iron and start hacking!
- Alternative-Z80 computer with a “new” design: **RC2014**  
WeChat: cstutorcs
- Build computer just from **logic gates**?

Assignment Project Exam Help



Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



## Further Reading

程序代写代做 CS 编程辅导

Essentials of Computer Organis



Architecture (Null) - Ch. 1.9

Essentials of Computer Organisation and Architecture (Null) - Ch. 1.10

WeChat: cstutorcs

Essentials of Computer Organisation and Architecture (Null) - Ch.1.11

Assignment Project Exam Help

Essentials of Computer Organisation and Architecture (Null) - Ch. 9

Email: tutorcs@163.com  
QQ: 749389476

GPU Topics Multicore and GPU Programming

<https://tutorcs.com>

## Questions

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>