Microprocessor & Microcontroller → used in OS

Upoma Madam  Tushar Sir

15.01.2024

Cortex M4 microprocessor (ARM) + Intel
- used in embedded applications

• Sudden exams MCQs

STM32 microcontroller

Book : Joseph Yiu
Reference Manual, Technical Manual পড়ে পড়ে বুঝতে হবে।
দেখে দেখে কাজ করতে হবে।

• ARM is a concept, a design principle. Various companies build processor based on this concept and its features.

• register is the internal memory of microprocessor. It is built in to the processor. Made of D-flip-flop because it is fast.

• CMOS technology - capacitor থাকে
time wastage for charge-discharge-recharge

• Memory mapping জানা very important.

• ARM instruction set & Thumb Instruction set.
set of 16 bit & 32 bit

Cortex M4 এ সবই thumb.
→ can be customized
→ has optional components like FPU, MMU → dedicated unit হলে faster. এটা হার্ডওয়্যার কাজ করা যাবে কিন্তু extra bus time, clock time use করে।

mandatory components : Debug & Trace features

22.01.2024

[UK Lecture 2]

## Microprocessor

- CISC and RISC

[Currently, combination of both]

- MIPS is a RISC processor.

Microprocessor is a chip. Every pin is a connection for different functionality. ভিতরে complicated circuitry.

DMA → এখানে processor কে use করিনা, free করে দেই। Because প্রতিটা কাজেই processor এর intervention লাগে।

- transistor → নানা gates → ALU → a component inside processor. various algorithms are also used in designing circuit inside processor.

- প্রতিটা input board → adapter → motherboard & processor.
- Lab থেকে mid/final এ question হবে।

- ALU never communicates with memory directly. ALU communicates with registers.

process: a program is in execution/running state → lines of instruction

↓

it is using some resource → memory

[RAM এ আমার আগে কখনো execute হয়না]

↓

কোন space empty + address calculation হয় + reconfiguration + repla

- আমি double click করলে সেটা hard disk থেকে RAM এ আসে

- এগুলো operating system handles using control unit and registers (e.g. address register, program counter)

- calculation straight-away হয়না, Different mechanisms for 32-bit, 64-bit processor.

  ↓

  $2^{64}$ ~~bit~~ addressing করতে পারে।

  At a time, 64 bit data transfer internally.

  Memory block 64 bit.

  word size is 64, ALU size is 64. ALU can process 64-bit data at a time

- একটা code এর নানা part distributed থাকতে পারে in memory. linker লাগে।

- processor নানা process কে serve করে, এত fast switch করে যে user বুঝতে পারেনা।

- একটা architecture family তে লেখা program is portable in all compatible processors. কোন processor doesn't matter.

  - Intel তাদের product কাউকে দেয়না।
  - ARM is just a concept.

## Assemby Language

opcode operand1, operand2, ...; Comments

↓

usually আগে destination

- এই slide এ যেই example: Intel case. 2 operand instruction set.
- ARM এ 3 operands allowed.
- LDR is an instruction of ARM architecture.
- Third bracket denotes indirect addressing. এই address এর value কে আনতে হবে।

microprocessor specs এর bus → internal bus. External bus অন্যান্য to communicate with peripheral devices.

Special Purpose microprocessor example: GPU

29.01.2024

- History of CISC. Microprocessor
- কোন processor থেকে peripheral communication start হলো?
  → Intel 8085 (because of I/O ports)
- Concept of virtual memory start from Intel 80386.
  2 bit multiplexing করে 20-bit address bus
- Intel 8086 এ pipelining concept.
  relies on
# superscalar architecture
- more transistor → more powerful
- 80386 এ যেই feature গুলা লেখা সেগুলো ভালো করে পড়ব.
- virtual memory → hard disk এ swap space create করি, swap
  -ping is so fast যে user বুঝতে পারেনা.
  - requires address translation → address translation register.
  - multitasking → scheduling
  - protection mechanism - we cannot access the memory যেখানে OS আছে. PC and RA থেকে calculated address যদি invalid হয় → error signal.

Pentium

superscalar → multiple instruction pipeline
→ overlapping concept. speed আরো বেশি.

- dual integer processor supports parallelism + superscalar technology
  → concept of coprocessor.

Imp

- GPU is a coprocessor.
- NOT instruction pipelining. ~~this is~~ co-processors are independent.

- Virtual memory is important for exam.
- branch prediction can help us decide which data to purge for virtual memory.

Q. Learn about cores.

problems:
① synchronization ② critical problem

① Core uses shared memory.

- Core এর সাথে কিছু private memory, কিছু register, কিছু memory allocation থাকে।

② CSMACD
↳ LAN এর communication tech

voltage level দেখে বুঝা যায় ওই channel দিয়ে ওই সময় data যাচ্ছে কিনা। যদি active থাকে তাহলে তখন ওখানে transmission ~~হয়~~ করেনা। certain random time পর আবার sense করা হয়। channel free থাকলে data পাঠানো শুরু করে।

- But ~~but~~ multiple device একই time এ sense করে একই time এ transmit করলে collision. → সেটা detect করে transmission থামাবে।

Q. RISC আসছে কেন?

- Almost any new development is based on requirements of American Defense. তাদের research institute → DARPA

Solaris OS → Sun machine

• superscalar, pipelining belonged to RISC. But currently hybrid, CISC এর ভিতরেও RISC এর কিছু concept চলে আসে।

• RISC is load and store architecture.

• Not in CISC. CISC এ নানা instruction এ কখনো কাজ PLUS memory access এভাবে কাজ করা possible.

• RISC এ 4-5টি specific ছাড়া সব general purpose registers. CISC এ সব predefined.

31.01.2024

64 pin microcontroller. - 8 ports
144 pin microprocessor. (?)

• MCQ তে history থেকেও আসবে।

Ulka-semi → Bangladeshi owner, semiconductor company.
Iphone 11/12 এ ওদের design করা chip আছে।

• Fabrication industry is very expensive. BD companies mainly design পর্যন্ত করে।

IP licensing - Intellectual Property Licensing.

• ARM এর দুর্বলতা cloud এ।

ARM Architecture

- সবার আগে একটা model দেয়। যেখানে instruction sets interrupt, exception, addressing modes, memory mapping সব থাকে।

• একেকটা ARM version and তাদের example নিজেরা দেখে নিব।

আমরা ARM version 7 পড়ছি।

A series : application

R series : Realtime

M series : Microcontroller

SecurCore series (SC) : High security applications

সবগুলোরই fundamental features → main model

* Mode দুইটা : ARM (32 bit), Thumb (16 bit) → switch করা যায় betn modes.

Embedded systems এ power efficiency, low heat dissipation, etc. important.

• Thumb mode → 16 bits use হয় instead of 32
↓
less flip flops, less power consumption, lower silicon cost.

• Thumb also has 32 bit instruction for 64 bit processors → Thumb2

• Von Neumann architecture.

~~Pg 9 → Hard~~

DSP : Digital Signal Processing

কিছু component optional, কিছু mandatory.

• Cortex M3, M4 similar. M4 just has DSP extensions and FPU extra.

• MAC - multiplication and Accumulation.
একটা instruction এই দুইটা কাজ হয়ে যায় → dual

SISD বাদে বাকিগুলো used in parallel processing/computing.

Q. Risc need more memory than cisc to perform the same action. why?

→ RISC এ ~~As of~~ more instruction লাগবে।

# Cortex-M4 এর সব কাজ Thumb এ হয়না,
অনেক সময় Thumb এ compile করলে backward compatibility থাকেনা, so Arm এ করতে হয়।

- Thumb 2 is not for normal operation, rather for exception handling, special functions like access to coprocessors (e.g. FPU, MMU), DSP operation.

- আগে data bus and address bus multiplexed ছিল, পরে individual separate করা হয়েছে, why?

# Exceptions, interrupts very important for exam.
NVIC → Interrupt Handler [Nested Vector Interrupt Controller]
- debugging support from hardware.

Barrel shifter
- SMUL - signed multiplication, B → Byte wise, T → Thumb

weekend : task
tuesday : যেই কাজ lab এ দেখাতে হবে

Operation Mode
- Thread Mode (general)
- Handler Mode (superuser) → যে reset করে। reset থেকে বের হয়
  ↳ when there is an exception, interrupt
  → সেসব case এ privilege, operation options বেড়ে
- SVC - Supervisor call (is it similar to sudo?)

Exception modes + System → privileged mode এ use করতে হয়।

Q. MCQ হতে পারে যে কোন সময়, কী হলে ARM কোন state এ থাকবে? [কোন mode]

privilege mode থেকে privilege mode এ movement এর কারণ হতে পারে priority of exceptions → low priority, high priority, etc.

[Transition Diagram] → slide 21

Q. Exam এ scenario থাকবে। Then বুঝাতে হবে processor কোন state এ যাবে? Lab Part

11/02/2024

- Interrupt হলে operating system এর state change হয়।

# preamption

Interrupt এর priority and process ID always higher থাকে।

- hardware level এ NVIC থাকার ফলে mode changing, switching etc. লাগেনা and অনেক faster কাজ হয়।

MPU

- user কে একজন অন্যজন এর থেকে protect, যেন কোনো overlap না থাকে।
- OS কে protect করা

NVIC

8 bit → we can handle $2^8$ (→ 256) interrupts.

Of these, 1 to 240 are external interrupts

The remaining 16 are defined interrupts

- tail-chaining → switching করবনা, overhead কমায়

# synchronization এর সাথে sleep এর interrelation.

atomic instruction - যেই instruction এর মাঝখানে কোনো interrupt ঢুকতে পারবেনা।

sleep, wakeup are atomic instructions.

এগুলোকে চাইলে NVIC এ configure করতে পারব,

- configurable option বাদে সবগুলোই mandatory.
- Microprocessor lab এ আমরা Code + SRAM part এ কাজ করব.

37 registers for ARM architecture. এখানে 16টা

[not specifically Cortex M4]

- একই register can function for both privileged user and normal user.

R13 → MSP (in privileged mode) → OS Kernel or Exception

→ PSP (in unprivileged mode) → user stack কে handle করবে.

- PC is auto increment. PC = PC + 4 bytes

special register - current operation এর status রাখে।

- ISR-Number → এই bit গুলো NVIC configure করতে use হয় APSR and IPSR পড়ব,

tuesday exam → slide 22 of today + GK DU

9/02/2024

- Final Exam এ each set এ 4টা question. কোনো একটায় max. marks 5/6.

- Fast Interrupt = অন্য interrupt এর চেয়ে priority বেশি।

Q. কোনো scenario/code অনুযায়ী register এর কোন bit set হবে এগুলো আসতে পারে।

Figure 6: এই 16টা defined interrupt. বাকিগুলো আমরা নিজের design করি। → number গুলো priority ও বুঝায়।

- EPSR to interrupt LDM (Load Multiple), STM

Mask → Disable.
i.e. Program চললে ওই interrupt গুলো কাজ করবেনা। কিছু interrupt আছে non-maskable, যেগুলোর priority বেশি।

- PRIMASK → time critical কিছু যেখানে আমি কোনো interrupt কে ঢুকতে দিবনা। User level এ।

- FAULTMASK → কোনো error/fault (e.g. divide by zero) handle করার সময় অন্য interrupt কে ঢুকতে দিবনা। fault handling is privileged mode, error is an exception in itself. অন্য exceptions ঢুকতে দিবনা।

- Interrupt service Routine.
কত number interrupt + memory location

- Reboot is an example of NMI.

06/03/2024

- Data Processing / PSR Transfer এগুলো incourse এ থাকবে।

32 bit কে নানা segment এ ভাগ করে রাখি।

- 28 - 31 4 bit → condition
  (ARM এর instruction conditional & unconditional হয়)
  - unconditional হলে ~~[illegible]~~ → 1110 থাকবে।
- 00 → reserve [26 - 27]
  - future এ অন্য কিছু তে utilize করা হবে।

- 25 → immediate → I = 0 means normal, 1 মানে immediate
- 20 - 24 → funct } opcode
- 24th bit → Flag set হবে কিনা
  Rn → ~~R0 (source)~~ R1 (1st operand)
  Rd → R0 (destination)  ADD R0, R1, R2
  operand 2 → R2 [0 - 11]

- | Shift | Rm |
  - Shift → any shift applied to Rm
  - Rm → 2nd operand register → nested operation এর ক্ষেত্রে এটা utilize করতে পারে since 12 bit. / যখন immediate নাই।

1st variant

| 11...8 | 7 | 6 | 5 4 | 3...0 |
|---|---|---|---|---|
| Rs | 0 | Sh | 1 | Rm |

→ register এর content দিয়ে shifting হবে। [Rs এর content দিয়ে Rm shift হবে]

| Sh | |
|---|---|
| 00 | LSL |
| 01 | LSR |
| 10 | ASR |
| 11 | ROR (rotation) → right |

- 4th bit = 0 (shift by a constant)
  = 1 (shift by a register)

- 7th bit → 0 for register
  → constant এর ক্ষেত্রে use হবে।

2 | 42
2 | 21 — 0
2 | 10 — 1
2 | 5 — 0
2 | 2 — 1
2 | 1 — 0
0 — 1

2nd variant

| 11....7 | 6 | 5 | 4 | 3:0 |
|---|---|---|---|---|
| shamt | sh | 0 | | Rm |

↓ shift by constant value.

For immediate,

example of immediate addressing mode

| 11...8 | 7...0 |
|---|---|
| Rotate | Imm |
| 0000 | |

e.g. MOV R5, #6

↓ 8 bit এখানে 256 পর্যন্ত। #256 পর্যন্ত normally handle করতে পারব। এরচেয়ে বেশির জন্য bit rotation necessary হবে। Exam এর পর details পড়ব।

(1) ADD R1, R2, R3

| cond | I | opcode | S | R2 | R1 | R3 |
|---|---|---|---|---|---|---|
| 1110 | 00 | 0 | 0100 | 0 | 0010 | 0001 | 00000000 0011 |

(2) ADD R0, R1, #42

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1110 | 00 | 1 | 0100 | 0 | 0001 | 0000 | 0000 | 00101010 |
| | | | | | | | ১০২ | #42 |

ADDS R5, R7, # 2

| 1110 | 00 | 1 | 0100 | 1 | 0111 | 0101 | ..... 10 |
|---|---|---|---|---|---|---|---|

→ 0xE2975002

SUBEQ

| 0000 | 00 | 0 | 0010 | 0 | 0010 | 0001 | 0000 0000 0011 |
|---|---|---|---|---|---|---|---|

→ exam এ এগুলোর equivalent hex code লিখতে হবে। OR hex code দেওয়া থাকবে, বলতে হবে কী instruction.

LSL R0, R0, #7 . যেকোনো shifting 1101 (MOV এর 3 same)

not immediate. R0 এ কাজ করে R0 তে রাখবে।

| 1110 | 00 | 0 | 1101 | 0 | 0000 | 0000 | 00111 | 00 | 0 | 1001 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | S | Rn | Rd | | #7 | sh | shift by constant | Rm |

operand 2

- basically shift করে move.

→ MOV & shift এর hex code same, আমাদের ভিতরের গিনিয়া দেখে বুঝার কোনটা হচ্ছে।

→ এখানে কোনো lsl operand নাই।

MOVS R0, #0

10 ASR

11 ROR (rotation)

MOVS R0, #9

| | | | | | Rn | Rd | #9 |
|---|---|---|---|---|---|---|---|
| 1110 | 00 | 1 | 1101 | 1 | 0000 | 0000 | 0000 00001001 |

ADD R0, R1, R2 LSL R3

| | | | | | R1 | R0 | R3 | LSL | R2 |
|---|---|---|---|---|---|---|---|---|---|
| 1110 | 00 | 0 | 0100 | 0 | 0001 | 0000 | 0011 | 0 00 | 1 0010 |

12 bit