**Microprocessor and Computer Architecture**

**Sub Code: UE18CS253 Sem: IV**

**Model Question Paper and Solution**

**Que1a) Write the key features used of RISC architecture in an ARM Processor. Explain. (6)**

**Solution:**

i. Load Store Architecture

ii. 3 -address instruction

iii. Fixed length 32 bit instructions.

**Que1b) Explain the following Instructions AND, ORR, EOR, EIC (4)**

**Solution:**

* AND r0, r1, r2 ;r0 := r1 AND r2
* ORR r0, r1, r2 ;r0 := r1 OR r2
* EOR r0, r1, r2 ;r0 := r1 XOR r2
* BIC r0, r1, r2 ;r0 := r1 AND (NOT r2), bit clear

**Que1c) The byte addressable memory system starts from location with an address (6)**

**0x00306090. The system uses little endian storage. Show the memory allocation**

**for the following data items declared in an ARM memory organization.**

**i. .byte 20, 50, 38, 45**

**ii. .halfword 4567**

**iii. .word 78650A09**

**Solution**:

i. 0x00306090 20, 0x00306091 50, 0x00306092 38, 0x00306092 45

ii. 0x00306090 67, 0x00306090 45

iii. 0x00306090 09, 0x00306091 0A, 0x00306092 65, 0x00306092 78

**Que1d) What is the difference between computer organization and computer architecture. (4)**

**Solution:**

**Architecture** describes about the User's view of the computer. The instruction set,

Visible registers, memory management table structures and exception handling

model is discussed.

**Organization** describes about the user-indivisible implementation of the architecture.

The pipeline structure, transparent cache, table-walking hardware and TLBare the accepts

of organization.

**Que2a) Write the instruction sequence for the following statements using ARM instruction set. (06)**

**i. IF ( X < 0 ) (X= X >>2)**

**ELSE X=X<<2**

Answer: assume X in Reg R0.

CMP R0, #0

MOV RO, RO, LSR #2

B L1

MOV RO, RO, LSL #2

L1: SWI 0X11

**ii. IF ( A == B ) { X = X + Y \* Z } ELSE { X = X + Y }**

Answer: Assume X is R0, Y is R1 and Z is R2. A & B are R3 and R4 respectively.

CMP R3 , R4

BEQ L1

ADD R0, R0, R1

B L2

L1: MLA R0, R0, R1, R2

L2: SW 0X11

**Que2b) Write ARM assembly programme to generate Fibonacci series. (06)**

Fibonacci series

.TEXT

MOV R0,#6

LDR R1,=A

MOV R2,#0

MOV R3,#1

STR R2,[R1],#4

STR R3,[R1],#4

LOOP1:ADD R4,R2,R3

STR R4,[R1],#4

MOV R2,R3

MOV R3,R4

SUBS R0,R0,#1

BNE LOOP1

SWI 0X011

.DATA

A:.WORD 0X00

**Que2c)** **What ARM instructions does this represent? { Consider 14-AL, 00- EQ, 04- ADD} (04)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Condition** | **F**  **(format)** | **I**  **(immediate)** | **OPCODE** | **S**  **(Set cond Code)** | **Rn** | **Rd** | **Operand2** |
| **14** | **0** | **0** | **4** | **0** | **0** | **1** | **2** |
|  | | | | | | | |
| **00** | **0** | **1** | **4** | **1** | **0** | **1** | **#2A** |
|  | | | | | | | |

**ANS: ADD R1, R0, R2**

**ADDEQS R1, R0, #2A**

**Que2d) How are nested subroutines executed? Explain. (04)**

**Solution:**

The Subroutine Call instruction is done by instruction - BL addr . Branch & Link to Destination.

Return from the subroutine is by the instruction - MOV PC, R14. Where R14 is the link register which carries the return address. Since only one link register is provided, the contents of R14 register will be lost if it is not saved. This is taken care by pushing the current R14 contents onto the stack and restore from the top of the stack when a return is called.

**Que3a) What is pipelined processor? Mention types of Pipeline hazards. (04)**

Solution : The Processor which overlaps instruction execution is called pipeline.

Types of Hazards: Structural Hazard, Data hazard and Control Hazard

**Que3b) In 5 stage pipeline, in which each stage takes 1 clock cycles. Assuming all memory references are hit. For the given lines of code calculate number of clock cycles requires for non-pipelined and pipelined processor. What is the speedup achieved because of pipeline (8)**

**Add 2 integer Array**

LDR R4,#400

L1: LDR R1, 0(R4)

LDR R2,400 (R4)

ADD R3,R1,R2

STR R3,0(R4)

SUBs R4,R4,#4

BNE L1

**Solution:**

1. Total CC = (Initial instruction + (Number of instructions in the loop L1) x number of loop cycles) x number of clock cycles / instruction (CPI)

= (1 + ( 6 ) x 400/4 ) x 5 CC = 3005 CC

1. Number of cycles in the loop = 15 CC

Number of clock cycles for segment execution on pipelined processor =

1 CC(IF stage of the initial instruction) + (Number of clock cycles in the loop L1) x Number of loop cycles = 1 + 15 x 400/4 = 1501 CC

Speedup of the pipelined processor comparing with non-pipelined processor =

Number of Clock cycles for the segment execution on non-pipelined processor / Number of Clock cycles for the segment execution on simple pipelined processor =

= 3005 CC / 1501 = 2 times.

**Que 3c) The pipelined processor has the following stages: Instruction Fetch (IF), Instruction Decode (ID), Instruction Execute (IE), write back (WB) and Write register (WR), The IF, ID, WB and WR stages take 1 clock cycle each for every instruction. Consider a sequence of 100 instructions. In the IE stage, 40 instructions take 3 clock cycles each, 35 instructions take 2 clock cycles each, and the remaining 25 instructions take 1 clock cycle each. Assume that there are no data hazards and no control hazards.**

**What is the number of clock cycles required for completion of execution of the sequence of instruction? (4)**

Solution: 219 Clock cycles

**Que 3d) For the purpose of solving a given application problem, you benchmark a program on two computer systems. On system A, the object code executed 80 million Arithmetic Logic Unit operations (ALU ops), 40 million load instructions, and 25 million branch instructions. On system B, the object code executed 50 million ALU ops, 50 million loads, and 40 million branch instructions. In both systems, each ALU op takes 1 clock cycles, each load takes 3 clock cycles, and each branch takes 5 clock cycles. Compute the relative frequency of occurrence of each type of instruction executed in both systems. (4)**

Solution:

Alu Output

A: 80/145 = 0.55 B: 50/140 = 0.36

Loads

A: 80/145 = 0.28 B: 50/140 = 0.36

Branches

A: 25/145 = 0.17 B: 40/140 = 0.28

**Que 4a) In a direct mapped cache organization, determine the number of bits required to**

**recognize a byte of a 128MB cache memory which has a Set that is consisting of 32 lines. Each line has 64 bytes. Also determine the number of bits required to reach a Tag, Set, Line, and Byte and write how many bytes are copied when data is loaded into the cache. (6)**

Solution:

Bits required**:**

Bytes are represented by using - 6 bits.

Lines are reached by decoding - 5 bits.

Sets # of =Capacity of the Memory cache /( # of bytes X #of lines)

128 x 1024 x 1024 bytes / 64 x 32 = 65536 sets requires 16 bits to decode.

Therefore 5 bit TAG + 16 bits SET + 5 bits LINE and 6 bits BYTE.

**Que4b) Explain briefly the two principal approaches to memory management. 06**

Solution:

i. Segmented memory management

ii. Paging Memory Management.

**Que4c) On what basis cache optimization is derived? Mention the six basic cache optimization techniques and explain. 08**

Solution:

**Average access time = Hit time + Miss rate x Miss penalty**

* Six “easy” ways to improve above equation
* Reducing miss rate:

1. Larger block size (compulsory misses)

2. Larger cache size (capacity misses)

3. Higher associativity (conflict misses)

* Reducing miss penalty:

4. Multilevel caches

5. Prioritize reads over writes

* Reducing hit time:

6. Avoiding address translation

**Que5a) Consider the following program: Find the ILP Factor. Assume each takes 1 Clock Cycles ( 4)**

**1. e = a + b**

**2. f = c + d**

**3. g = e \* f**

**Solution:** If we assume that each operation can be completed in one unit of time then these three instructions can be completed in a total of two units of time, giving an ILP factor of 3/2; which means 3/2 = 1.5 greater than without ILP.

**Que5b) What is the difference between Scalar and Super scalar Pipeline (8)**

Scalar Pipeline is divided into various pipeline stages:

• Fetch Decode Execute Memory Access Write Result



• Superscalar Pipelines includes:

• **Parallel Pipelines**

• Wide pipelines

• Multiple instructions execution per cycle

• **Diversified Pipelines / Decoupled Pipelines**

• Multiple functional units

• Mix of different functional units

• **Dynamic Pipelines**

• Out of order execution

• Distributed functional units



**Que 5c) What are the limits of loop unroll (5)**

Amount of loop overhead amortized with each unroll

• Unroll 4 times , 2 out 14 cycles are overhead , 0.5 cycles per iteration

• Unroll 8 times , 0.25 cycles per iteration

• Growth in code size

• Larger loops, code sizes growth will be a concern

• Large code size may increase instruction cache miss rate

• Potential shortfall in registers that is created by aggressive unrolling and

Scheduling strategy

• Register pressure

• Scheduling code to increase ILP causes the number of live values to increase, thus

generates shortage of registers

**Que 5d) What are the advantages of multithreading (3)**

If a thread can not use all the computing resources of the CPU (because instructions depend on each other's result), running another thread permits to not leave these idle. If several threads work on the same set of data, they can actually share its caching, leading to better cache usage or synchronization on its values.

  If a thread gets a lot of cache misses, the other thread(s) can continue, taking advantage of the unused computing resources, which thus can lead to faster overall execution, as these resources would have been idle if only a single thread was executed