

CT1 Übungsaufgaben Architektur

1. Nennen Sie die Hauptbestandteile der M0 CPU und erklären Sie ihre Funktion.

Core Registers	16 32-bit Register
ALU	Rechenwerk für logische und arithmetische Operationen
Flags	Zeigt den internen Status des Prozessors an.
Instruction Register	Register, das den Maschinencode (Opcode) enthält, der im Moment ausgeführt wird.
Bus Interface	Schnittstelle zum externen System Bus; Umsetzung des internen Busses auf den externen Bus

2. Ein Prozessor arbeitet eine Liste von Befehlen in einer vorgegebenen Reihenfolge ab. Finden Sie Analogien aus dem täglichen Leben.

Bäcker/Koch, der nach einem Rezept arbeitet
Musiker, der nach Noten spielt
Pilot, der vor dem Start eine Checkliste abarbeitet

3. Wie viele Bytes können mit einer 8-bit Speicheradresse adressiert werden? Wie viele mit einer 16-bit Adresse? Wie viele mit einer 32-bit Adresse?

8 bit → 256 Bytes
16 bit → 64 KBytes = 65'536 Bytes
32 bit → 4 GBytes = 4'294'967'296 Bytes

4. Nennen Sie 3 Instruktionstypen der M0 CPU.

Data transfer
Data processing
Flow control

5. What is the function of the following registers?

a. PC

Program counter: Points to the address where the instructions will next be read

b. SP

Stack pointer: Points to the memory addresses where the elements are written/read from the stack

c. LR

Link Register: Used to keep track of the positions where to jump back (e.g. routines)

6. Why is the PC initialized to a defined value at reset? (Although other CPU registers may have an undefined content)

So that fetching of the first instruction can always start at the same (known and predictable) place.

7. Name the different parts of an assembly instruction.

Label, mnemonic, operands, comment

8. What is a memory map? What is it used for?

It is a graphical layout (map) showing the addresses and sizes of elements that communicate with the CPU (memories, Inputs, Outputs)

The memory map helps users to know where each element is (e.g. when writing the appropriate drivers)

9. How many byte positions can be addressed by the M0? Which positions in the memory map need to be occupied? Explain your answer.

The M0 has a 32-Bit address bus. Therefore, it can address 4 GByte = 2^{32} Bytes

- Positions needed for the initialization of the processor (at Boot) must be covered by the proper elements (memory). Otherwise, there will be no correct start.

10. Explain the following terms

a. Fetch

Get the instruction from code memory

b. Execute

Do what the instructions say

c. Word

A 32-bit memory unit (for the M0)

d. Half-word

A 16-bit memory unit

e. Little endian

A multi-byte representation where the LSByte is at the lower address

f. Big endian

A multi-byte representation where the MSByte is at the lower address

g. Word Alignment

The address of the multi-byte element is a multiple of the word length (4 for the M0)

11. A program (code in C) has variables represented as below in the memory map. Determine the decimal values of the variables Var1 Var5. Assume Little Endian representation.

Address	Byte content (decimal, hex, binary)	Variable
0x3000'0005	0x92	Var5 (char)
0x3000'0004	0x03	Var2 (unsigned char)
0x3000'0003	0xFF	
0x3000'0002	0x54	
0x3000'0001	34	
0x3000'0000	0xA2	Var1 (integer)
0x2FFF'FFFF	0x82	
0x2FFF'FFFE	0xA3	
0x2FFF'FFFD	10101101(binary)	
0x2FFF'FFFC	0x65	Var3 (unsigned integer)
0x2FFF'FFFB	25	
0x2FFF'FFFA	213	
0x2FFF'FFF9	01100010 (binary)	
0x2FFF'FFF8	0xE2	Var4 (short)
0x2FFF'FFF7	0x45	

Var1 =

Var1 : word (4 bytes) (0x FF5422A2) Var1 = -11263326 (it is 32-bit signed)

Var2 =

Var2 = 03 (it is 8-bit unsigned)

Var3 =

Var3 = 0x82A38D65 Var3 = 2191756645 (it is 32-bit unsigned!!)

Var4 =

Var4 = 0x62E2 var4 = +25314 (it is 16-bit signed)

Var5 =

Var5 =0x92 = -110 (it is 8-bit signed)

How would the same variable values be stored on a Big Endian platform? Fill in the table.

Address	Byte content (decimal, hex, binary)	Variable
0x3000'0005	0x92	Var5 (char)
0x3000'0004	0x03	Var2 (unsigned char)
0x3000'0003	0xA2	
0x3000'0002	34	
0x3000'0001	0x54	
0x3000'0000	0xFF	Var1 (integer)
0x2FFF'FFFF	0x65	
0x2FFF'FFFE	10101101(binary)	
0x2FFF'FFFD	0xA3	
0x2FFF'FFFC	0x82	Var3 (unsigned integer)
0x2FFF'FFFB	25 (or 213)	
0x2FFF'FFFA	213 (or 25)	
0x2FFF'FFF9	0xE2	
0x2FFF'FFF8	01100010 (binary)	Var4 (short)
0x2FFF'FFF7	0x45	

We are not told if 25 / 213 form a unit or not. Therefore, it can be both ways.

12. Welche 3 Speicherbereiche können bei einem Programm unterschieden werden? Was wird in den einzelnen Bereichen abgelegt? In welchem Speichertyp können die Bereiche jeweils liegen?

CODE read-only → RAM oder ROM
Maschineninstruktionen, Konstanten

DATA read-write → RAM
globale Variablen, static Variablen und heap in C

STACK read-write → RAM
Prozeduraufrufe/ Übergabe von Parametern, Lokale Variablen

13. Bei der Ausführung eines Programmes werden die folgenden Speicherbereiche benutzt:

- Code: 0x20000000 bis 0x200001FF
- Daten: 0x20000200 bis 0x200002FF
- Stack: 0x20000300 bis 0x200003FF

Zeichnen Sie eine entsprechende Memory Map und zeichnen Sie die drei Bereiche ein. Beschriften Sie für jeden Bereich jeweils die erste und die letzte Adresse. Wie viele Speicherstellen enthält jeder der Bereiche?

