



## Memory

To demonstrate the ability to understand how memory circuits are constructed using RS and D flop-flops, how to compute maximum addressable memory sizes given word sizes and addressability information and the difference in memory layout for little endian and big endian machines

Read carefully and check all statements below you agree with or that are correct about your assignment submission.

- ☐ If something is not clear in this page I ask for help from instructor or TA during submission period.
- ☐ After submission I will get a tentative assessment based on my input, that may be changed later after re-evaluation by grader.
- ☐ My answers represent my own individual work.
- ☐ Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY.



1



- NOR gates are used to build the RS and D flop-flops. It is also possible to use NAND gates instead.
- On a sheet of paper draw a logic diagrams of a RS flip-flop with two inputs (R and S) and two outputs (Q and Q') using NAND gates.
- Write down the truth table of the above circuit.
- Submit the paper at the beginning of the 1st class next week.

☐ OK

1



- On the same sheet of paper draw a logic diagrams of a D flip-flop with two inputs (D and E) and two outputs (Q and Q') also using NAND gates.
- Write down the truth table of the above circuit.
- Submit the paper at the beginning of the 1st class next week.

☐ OK

1



- Determine the memory size according to the following memory configurations. Indicate the memory size in the units of bytes.

- Byte-addressable memory, 10 bits used for addressing.

- Word-addressable memory, Word is 2 byte long. 16 bits used for addressing.

- Word-addressable memory, Word is 4 byte long. 32 bits used for addressing.

★	1	🔍
---	---	---

- How the following variable is stored in memory for Big Endian?

```
char x = 187;
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.

- Same for Little Endian.

★	1	🔍
---	---	---

- How the following variable is stored in memory for Big Endian?

```
int x = 180;
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.

- Same for Little Endian.

★	1	🔍
---	---	---

- How the following variable is stored in memory for Big Endian?

```
int x = 0x3C2874A7;
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.

- Same for Little Endian.

★	1	🔍
---	---	---

- How the following variable is stored in memory for Big Endian?

```
short x = 0x82BA;
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.

- Same for Little Endian.

★	1	🔍
---	---	---

- How the following array is stored in memory for Big Endian?

```
unsigned int x[2] = { 0x9381DC7E, 0x274B1A9D };
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.

- Same for Little Endian.

★	1	🔍
---	---	---

- How the following null-terminated string is stored in memory for Big Endian?

```
char str[] = "ABC";
```

- Give the answer in hexadecimal system. Ex: 0C for 1 byte long data or 1AB0 for 2 byte long and so on.
- Also include the terminate byte 0 into the answer.

- Same for Little Endian.

★	1	🔍
---	---	---

- What is approximate number of hours you spent implementing this assignment?

- Indicate the specific portions of the assignment that gave you the most trouble

By signing this document you fully agree that all information

Responder sign:

provided therein is complete and true in all respects.