DT509G Lab 1: Digital Logic Circuits

Anton Söderlund, Marcus Nyström
20180921

Task 1

We set up the breadboard as instructed. We tried all 4 combinations with the code in Appendix 1 to see that NAND-gate works as intended by turning the LED on and off at one second intervals. Attached sketch and showed the lab instructor.

Task 2

We started by looking at the truth table for the half-adder.

A	B	Sum	Carry
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

It shows what output we should expect from any combination of input. We started with building an XOR-gate using four NAND-gates that produces the sum output. To produce the carry bit, we added the NOR-gate as shown in Figure 1.

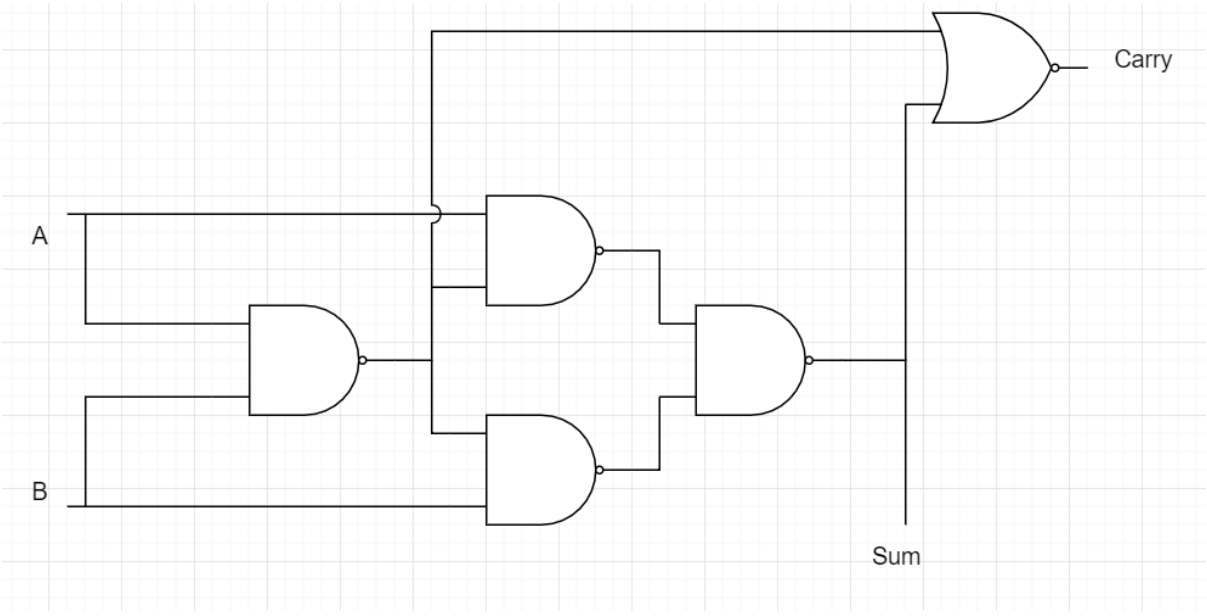

Figure 1: Schematics for half-adder.

We continued with building the circuit onto the breadboard corresponding with our schema, as can be seen in Figure 2. We then ran tests to confirm all combinations according to the truth table. See Appendix 2 for the code for the test.

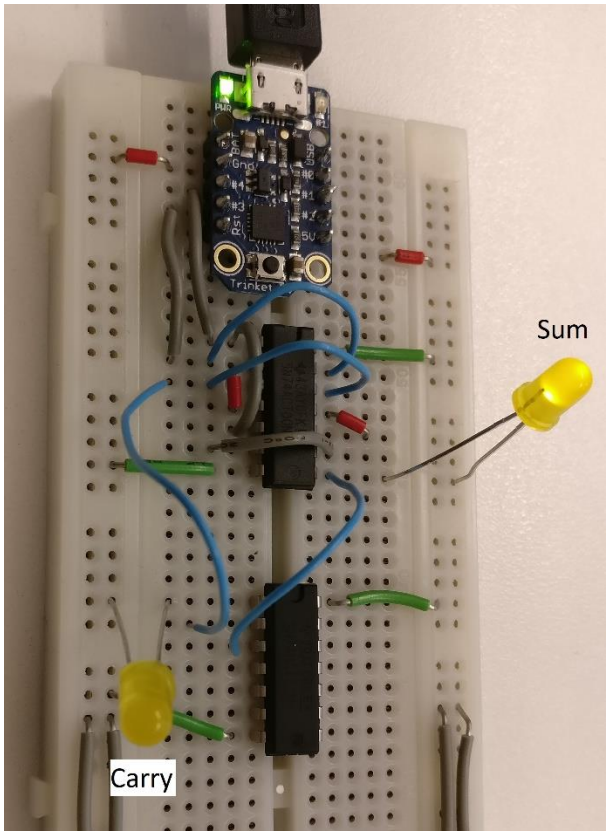

Figure 2: Half-adder using all NAND gates on a 7400 chip and one NOR gate on a 7402 chip.

Task 3

We started with copying the first circuit twice. Out of the five ICs we had five NOR-gates left in total. The OR-gate was then built by two NOR-gates in series, our implementation can be seen in Figure 3. We added more inputs in the sketch and tested the full-adder with two LED's, red for sum and yellow for carry. We tested all the combinations according to this truth table with the code shown in Appendix 3.

Full-adder

A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

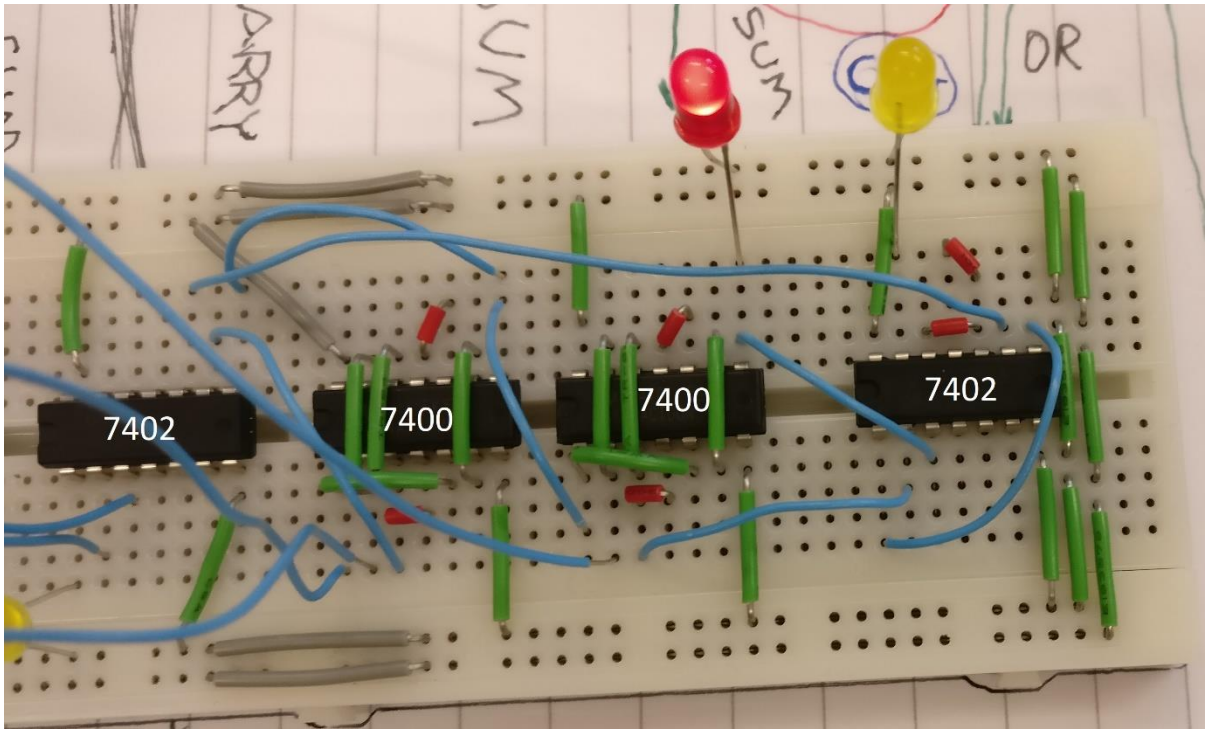

Figure 3: Full-adder implemented using two half-adders and an OR gate implemented using two NOR gates.

After we verified the full-adder worked correctly we chained the half-adder's carry output to the full-adder's carry input, the schematics can be seen in Figure 4. The truth table was derived from the two previous truth tables. A0, B0 represents the input to the half-adder, A1, B1 is the input to the full-adder, from the microcontroller.

Full-adder and 2-bit adder circuit

A0	B0	A1	B1	Sum0	C0	Sumout	Cout
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0
0	0	1	1	0	0	0	1
0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	0	1
1	1	1	0	0	1	0	1
1	1	1	1	0	1	1	1

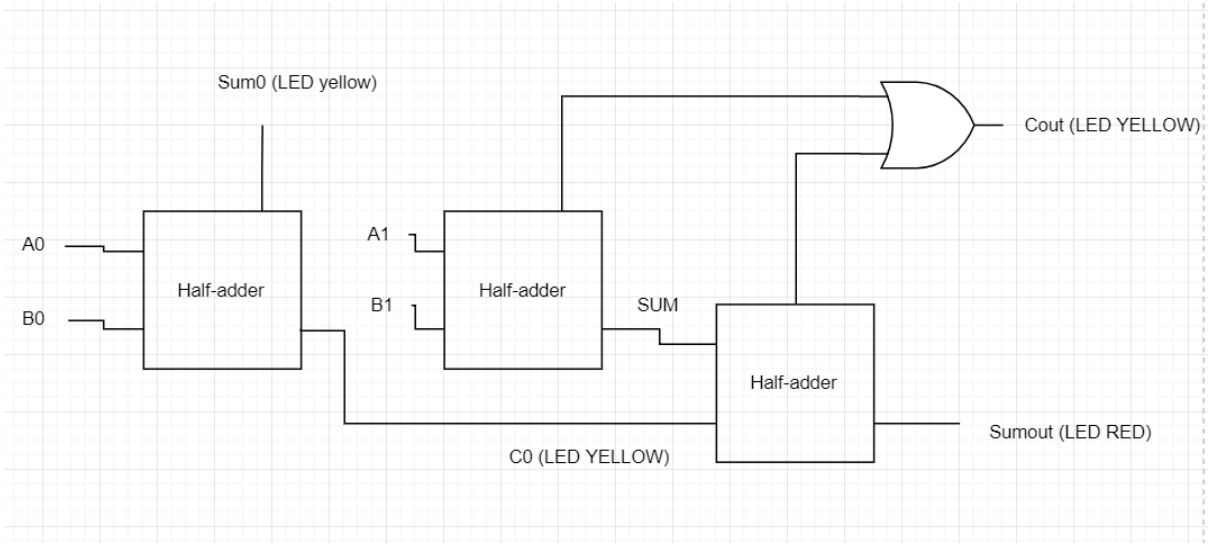

Figure 4: Schematics for the complete circuit.

The complete circuit, as can be seen in Figure 5, was demonstrated to the lab instructor using the code in Appendix 4, we manually changed the highs and lows of the inputs according to the instructor to show that the circuit worked properly at all input configurations.

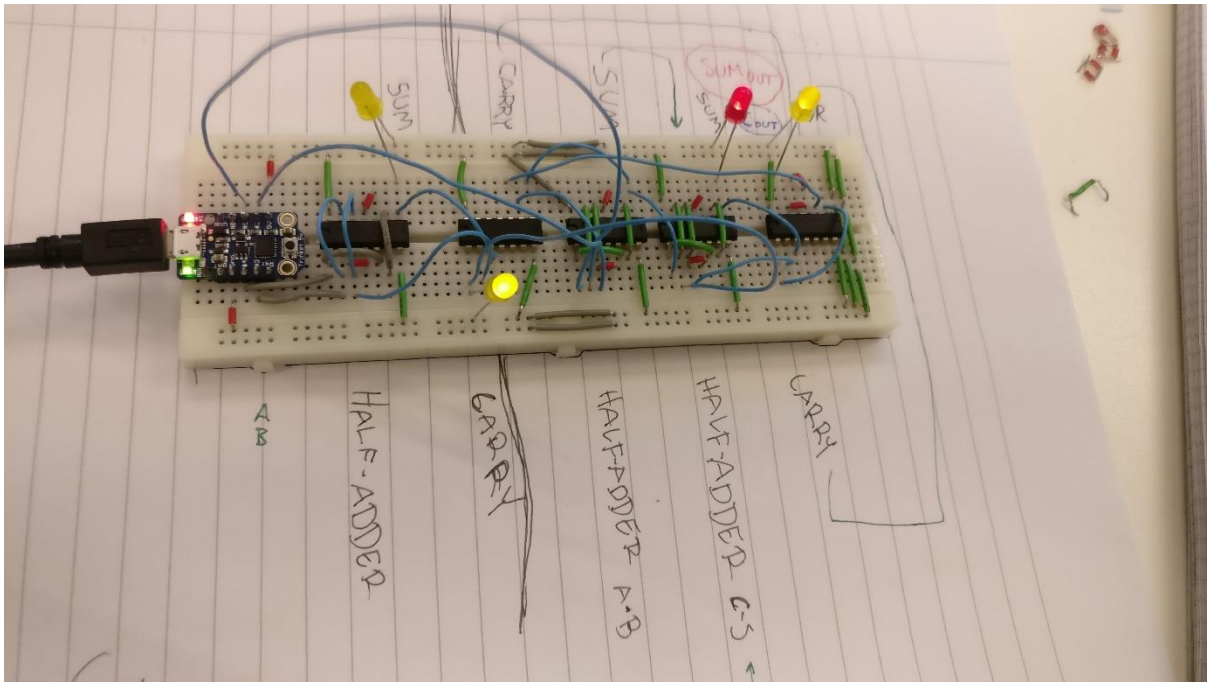

Figure 5: The complete circuit.

Appendices: Arduino Sketches

Appendix 1: Task 1

```
int pin3 = 3;
int pin4 = 4;
void setup() {
  // put your setup code here, to run once:
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
    digitalWrite(pin3, HIGH);
    digitalWrite(pin4, HIGH);
    delay(1000);
    digitalWrite(pin3, LOW);
    delay(1000);
    digitalWrite(pin3, HIGH);
    delay(1000);
    digitalWrite(pin4, LOW);
    delay(1000);
    digitalWrite(pin4, HIGH);
    delay(1000);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    delay(1000);

}
```

Appendix 2: Task 2

```
int pin3 = 3;
int pin4 = 4;
void setup() {
  // put your setup code here, to run once:
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);

  // TEST 1
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, HIGH);

  //TEST 2
  //digitalWrite(pin3, LOW);
  //digitalWrite(pin4, HIGH);

  //TEST 3
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, LOW);

  //TEST 4
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Appendix 3: Task 3, full-adder test

```
int pin3 = 3;
int pin4 = 4;
int cin = 2;

void setup() {
  // put your setup code here, to run once:
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(cin, OUTPUT);

  // TEST 1
  //digitalWrite(pin3, LOW);
  //digitalWrite(pin4, LOW);
  //digitalWrite(cin, LOW);

  //TEST 2
  //digitalWrite(pin3, LOW);
  //digitalWrite(pin4, LOW);
  //digitalWrite(cin, HIGH);

  //TEST 3
  //digitalWrite(pin3, LOW);
  //digitalWrite(pin4, HIGH);
  //digitalWrite(cin, LOW);

  //TEST 4
  //digitalWrite(pin3, LOW);
  //digitalWrite(pin4, HIGH);
  //digitalWrite(cin, HIGH);

  //TEST 5
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, LOW);
  //digitalWrite(cin, LOW);

  //TEST 6
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, LOW);
  //digitalWrite(cin, HIGH);

  //TEST 7
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, HIGH);
  //digitalWrite(cin, LOW);

  //TEST 8
  //digitalWrite(pin3, HIGH);
  //digitalWrite(pin4, HIGH);
  //digitalWrite(cin, HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Appendix 4: Task 3, complete circuit

```
int a0 = 3;
int b0 = 4;
int a1 = 1;
int b1 = 2;

void setup() {
  // put your setup code here, to run once:
  pinMode(a0, OUTPUT);
  pinMode(b0, OUTPUT);
  pinMode(a1, OUTPUT);
  pinMode(b1, OUTPUT);

  //TEST 1
  digitalWrite(a0, HIGH);
  digitalWrite(b0, HIGH);
  digitalWrite(a1, HIGH);
  digitalWrite(b1, HIGH);

}

void loop() {
  // put your main code here, to run repeatedly:

}
```