COMPUTER ORGANIZATION AND ARCHITECTURE

Lecture 5

Binary Arithmetic

Dr Victoria Mukami

INTRODUCTION

In the last lecture, we did a review of converting binary to understand what a computer goes through when converting a decimal, octal or hexadecimal system to binary. This week we focus on binary arithmetic. We will focus on the four basic calculations: Addition, Subtraction, Multiplication and Division.

Learning objectives

By the end of this topic, you should be able to:

1. Perform addition and subtraction operations with binary numbers
2. Perform multiplication and division operations with binary numbers
3. Understand the role of binary arithmetic

BINARY ADDITION

For us to understand how binary arithmetic works, we will first look at the addition that happens within the decimal number system. For instance, assuming one wanted to add 59+73 then the following would happen:

$$\begin{array}{rl} 1 & \\ 5\,9 & = 12 \\ +\;7\,3 & \\ \hline 1\,3\,2 & \end{array}$$

Within the decimal system, when adding you review the values. When we added 9 to 3, we got 12. 2 was dropped and a 1 was carried to the next part. It is the same in binary. For binary, we have some rules that we follow.

Sum	Result
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	1 0 (1 carry 1)

When working with 1 + 1 the result will be 11. However, the 1 will be dropped while the other 1 is carried to the next numbers. This is like decimal addition. Let us now work out some examples. We will ensure that we cover all the rules from the table during the examples.

Example 1

Add 10010 + 1001

```
  1 0 0 1 0
 +1 0 0 1
 ---------
  1 1 0 1 1
```

Example 2

Add 10101 + 1101

```
   1 0 1 0 1
 +   1 1 0 1
 -----------
 1 1 1 1 1 0
```

Following the rules 1 +1 is 1 carry 1. But 1 + 0 is 1. Hence the answers above.

Example 3

Add 10111 + 1001

```
   1 0 1 1 1
 +   1 0 0 1
 -----------
 1 0 0 0 0 0
```

From this example, we can see that When you add several 1's you are following the 1 + 1 rule. Let us now try adding three numbers.

Example 4

Add 1110111 + 101101 + 1101

```
   1 1 1 0 1 1 1          1 0 1 0 1 1 0 0
     1 0 1 1 0 1                  1 1 0 1
 ---------------          ---------------
 1 0 1 0 1 1 0 0          1 0 1 1 1 0 0 1
```

When you need to add binary numbers that have three different numbers, it helps to break them into two numbers. Then where you end up with three 1's it would help that you first work out the two 1’s and then add the third to the answer. Basically 1 + 1 + 1 you start with 1 + 1 = 1 0. So, the answer is 0 and carry 1. Then add the next 1 to the 0 and you get a 1.

BINARY SUBTRACTION

To best understand binary subtraction, we will first review how subtraction is done within the decimal system. Especially when you have to borrow some digits.

To best understand binary subtraction, we will first review how subtraction is done within the decimal system. Especially when you must borrow some digits.

$$\begin{array}{r} 10\ \ 10\ \ 10 \\ 4\cancel{5}\ \cancel{2}1\ \cancel{1}0\ 0 \\ -\ \ 7\ \ 8\ \ 9 \\ \hline 4\ \ 4\ \ 2\ \ 1 \end{array}$$

0 – 9 means that you borrow 10 from the previous number. The previous number is 1-8. When you borrow from the 1 you are left with 0. Then to subtract 0 from 8 you need to borrow from the previous number, The previous number is 2-7. When you borrow from 2 you are left with 1 and must borrow from 5. Hence the final answer is 4421. Assuming we had the following:

$$\begin{array}{r} 9\ \ \ \ \\ 9\ \ \cancel{10}\ \ 10 \\ \cancel{10}\ \ \ \ \ \ \ \\ 4\cancel{5}\ \ 0\ \ 0\ \ 0 \\ -\ \ 7\ \ 8\ \ 9 \\ \hline 4\ \ 3\ \ 1\ \ 1 \end{array}$$

Then you notice in the above example since 9 cannot borrow from 0 nor can 8 then 7 borrows from 5. 5 becomes a 4 and 7 gets a 10 above it. 8 borrows from the 10, which then reduces by 1. And this goes on until 9 gets a 10 above it and the subtraction is done. Now let us look at the rules of subtraction.

Subtract	Answer	Borrow
0 - 0	0	0
1 - 0	1	0
1 - 1	0	0
0 - 1	1	1

Just like in decimal when you borrow 10 from the previous number, in Binary you borrow 2 digits. In this case two 1's. When you have a 0 – 1 then you borrow from the previous number if it is a 1, If not you borrow from the next. When you borrow from a 1 then the remainder is a 0 and the number gets a 1 and 1. So as not to confuse you let us work on some examples.

Example 1

Subtract 11101 – 1001

$$\begin{array}{cccccc} & 1 & 1 & 1 & 0 & 1 \\ - & & 1 & 0 & 0 & 1 \\ \hline & 1 & 0 & 1 & 0 & 0 \end{array}$$

This is pretty much straightforward as it follows the first three rules. Let us now try one with the fourth rule.

Example 2

Subtract 11101 – 1011

$$\begin{array}{cccccc} & & & & 11 & \\ & 1 & 1 & \cancel{1}\,0 & 0 & 1 \\ - & & 1 & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 & 1 & 0 \end{array}$$

As seen in this example, the 0 borrows two digits from the previous 1. The third one remains as a 0 and the 0 gets 2 digits (1 1). So if we concentrate on the second part where we had 0-1 then when it gets two 1's, you have 1-0=1 then 1-1=0 then you drop the 1. Let's try another example that is a bit complex.

Example 3

Subtract 1001 –111

$$\begin{array}{ccccc} & 0 & 1\cancel{1} & 11 & \\ & \cancel{1} & 0 & 0 & 1 \\ - & & 1 & 1 & 1 \\ \hline & 0 & 0 & 1 & 0 \end{array}$$

This is interesting since the second digit cannot borrow from the third. Therefore, the third digit borrows from the fourth and receives two 1’s. The second digit borrows from the third, leaving the third with one 1 and the second gets two 1’s.

BINARY MULTIPLICATION

Multiplication is like long multiplication and the rules are like decimal, where 0 X 1 = 0 and 1 X 1 = 1. Let’s work on some examples.

Example 1

Multiply 1001 by 111

$$\begin{array}{rrrrrrr}
 & & & 1 & 0 & 0 & 1 \\
\times & & & & 1 & 1 & 1 \\
\hline
 & & & 1 & 0 & 0 & 1 \\
 & & 1 & 0 & 0 & 1 & \\
+ & 1 & 0 & 0 & 1 & & \\
\hline
1 & 1 & 1 & 1 & 1 & 1 &
\end{array}$$

We can see that as in long multiplication, each digit from the least significant in the second row is multiplied before we move to the next digit within the same row. Multiplication is truly straightforward.

BINARY DIVISION

Binary division works the same way that decimal long division works. The rule states that any number can only go into a specific number once. Let us do an example to understand this better.

Example 1

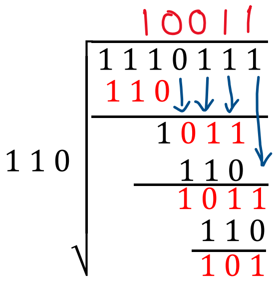
Divide 1110111 by 110

$$\begin{array}{r|l}
 & \ \ \ 1 \\
\hline
110 & 1110111 \\
 & 110 \\
\cline{2-2}
 & \ \ \ 1
\end{array}$$

The first step is arranging the question as shown above. We then divide for the first round. Here we realize that 110 can go once on 111. The remainder is 1. We add a 1 at the top since it went through once.

$$\begin{array}{r|l}
 & \ \ \ 100 \\
\hline
110 & 1110111 \\
 & 110\downarrow\downarrow\downarrow \\
\cline{2-2}
 & \ \ \ 1011
\end{array}$$

When we drop 0, 110 cannot go into 10. When we drop the 1, the number is still smaller than 110. We add a 0 on top. Next, we drop the 1.

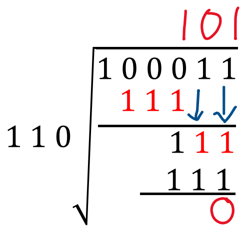


When we subtract 110, we add a 1 at the top, drop the next 1 and add a 1 at the top. When we finally finish our answer is <u>10011 rem 101</u>

Let us work out one more example.

Example 2

Divide 100011 by 111



Our final answer is <u>101.</u>

SUMMARY

During this lecture, we have reviewed binary arithmetic. Specifically, we have understood how we perform binary addition, subtraction, multiplication, and division. In the next step, we will learn about digital logic and circuits. Remember we learnt while computers understand binary, they communicate with IC's whose fundamental block is what we call logic gates. We will study this in the next lecture.

DISCUSSION TOPIC

After all the conversions, do you think computers would work on various binary arithmetic equations? If so, how would they work out something like the following:

$$11011 + 1101 - 111 \times 11100 \div 111$$

Would the computer use BODMAS?

REFERENCES

[1] G. Shelly and M. Vermaat, Discovering Computers — Fundamentals: Your Interactive Guide to the Digital World. Boston, MA: Course Technology, 2012.

[2] W. Stallings, Computer Organization and Architecture Designing for Performance. Hoboken, NJ: Pearson Education, Inc, 2016.

[3] A. Evans, K. Martin and A. Poatsy, Technology in Action. New York, NY: Pearson, 2020

[4] D. Morley and C. Parker, Understanding Computers: Today and Tomorrow. Boston, MA: Course Technology, 2017.