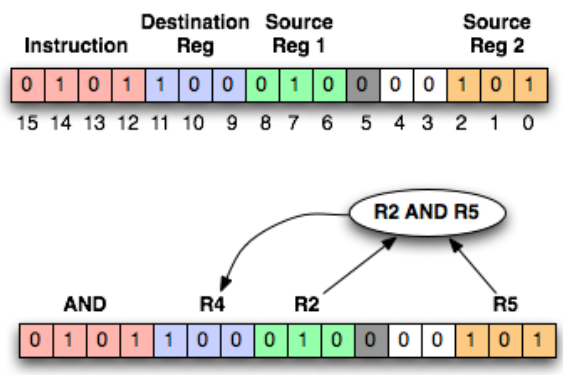
The AND Instruction

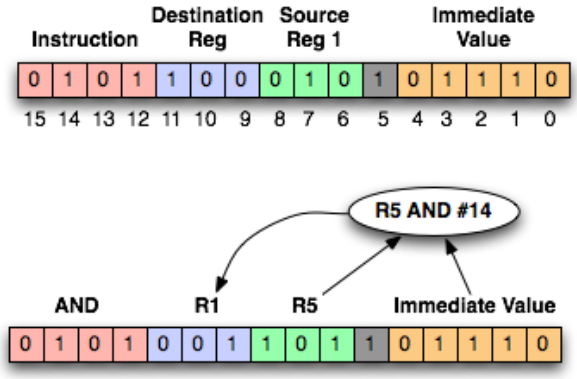
The High Level
The **AND** instruction is a type of **Arithmetic/Logic instruction instruction** that logically ANDs two values together and stores the result in a **destination register**. When you "AND" two bits together, the result is only 1 (i.e. true) if both of the bits are 1 (See Table 1 below).

AND	True	False
True	True	False
False	False	False

Table 1: Truth table for AND



(a) The AND Instruction – Flavor 1



(b) The AND Instruction – Flavor 2

Flavor 1 Breakdown:

- Bits [15-12] specify what instruction to execute (0101 is AND)
- Bits [11-9] specify which register to store the result in
- Bits [8-6] specify the left-hand side of the expression: **A AND B**
- Bit [5] specifies which form of AND is being used (in this case, the kind that ANDs two registers together)
- Bits [2-0] specify the right-hand side of the expression: **A AND B**

Flavor 2 Breakdown:

- Bits [15-12] specify what instruction to execute (0101 is AND)
- Bits [11-9] specify which register to store the result in
- Bits [8-6] specify the left-hand side of the expression: **A + B**
- Bit [5] specifies which form of ADD is being used (in this case, the kind that adds a register to an immediate value)
- Bits [4-0] specify the right-hand side of the expression **A + B** as an immediate value

The Examples!

```
AND R1, R2, R3      ; R1 <-- R2 AND R3
AND R4, R4, #0      ; R4 <-- R4 AND 0     (Protip: This sets R4 to 0)
AND R4, R4, #1      ; R4 <-- R4 AND 1     (Protip: Does not change the value of R4)
```

Pitfalls... (aka: Erroneous code makes baby dolphins cry)
The example below is erroneous. Please do NOT try to code this way!

```
AND R6, #10, R4     ; (ERROR: Immediate value must be the last operand)
AND R5, #2, #1      ; (ERROR: No flavor allows two immediate values)
```

The first example pitfall code above is incorrect because neither of the two flavors of AND allow for an immediate value to be the second operand given to the instruction.

The second example pitfall code above is incorrect because there are no forms of the AND instruction that allow two immediate values to be ANDed together. You must either put both values into registers and AND the registers together, or put one value into a register and use the second flavor of AND.