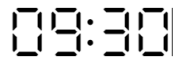
WHY BCD?

Suppose we want to store the time 09:30

09 ➔ 00001001

30 ➔ 00011110

09:30 ➔ 00001001 **: 0**0011110

To display 09:30 on a digital display:- 

Step1: Convert 0001001 to 09 (using powers of 2 i.e. 0*2^7 + 0*2^6 + 0*2^5 + 1*2^4 +…)
Step2: Separate the decimal digits 0 and 9
Step3: Display 0 then display 9
Step4: Convert 001110 to 30 (using powers of 2 i.e. 0*2^7 + 0*2^6 + 1*2^5 + 1*2^4 +…)
Step5: Separate the decimal digits 3 and 0
Step6: Display 3 then display 0
Step7: Repeat steps 1-6 whenever time changes.
There has got to be an easier way! =➔ BCD

Step1: Store the time 09:30 as 0000 1001 : 0011 0000
Step2: Display 0000 as 0, 1001 as 9, 0011 as 3 and 0000 as 0.
When we store each decimal digit separately in binary form,
that format is nothing but BCD.
(BCD = Binary Coded Decimal)
Step3: Whenever time changes only update required part of binary number

**Suppose we want to increment the hour i.e. change 09:30 to 10:30.**
But if we increment 9, it gives us 10, so 09:30 becomes 00:30
WHY?

- When we increment 09, we increment only 9 and not the zero, because 0 and 9 are stored separately.
- 9 becomes 10 and we store only one (the lower) digit.
- The upper 0 remains as it is. So we get 00:30.
- But we want 10:30

**Solution - BCD addition!**

**Example1:**

**8 + 7 = 15**
**1000**
**+ 0111**
**----------**
**1111**
**+ 0110**
**----------**
**1 0101**

**Example2:**
**9 + 8 = 17**
**1001**
**+ 1000**
**----------**
**1 0001**
**0110**
**----------**
**1 0111 = (17)bcd**

**89 + 79 = 168**
**1000 1001**
**+ 0111 1001**
**----------**
**1 0000 0010**
**0110 0110**
**----------**
**1 0110 1000 = 168**

**279 + 781 = 1060**
**0010 0111 1001**
**+ 0111 1000 0001**
**----------**
**1001 1111 1010**
**+ 0110 0110 0110**
**----------**
**1 0000 0110 0000**
**0001 0000 0110 0000 ➔ 1060**