|  |  |  |  |
| --- | --- | --- | --- |
| Date: | **05-08-2020** | Name: | **Varun G Shetty** |
| Course: | **Coursera** | USN: | **4AL17EC093** |
| Topic: | VLSI CAD part I | Semester & Section: | **6th & ‘B’** |
| GitHub Repository: | **Varunshetty4** |  |  |

**Report-**

Advanced Boolean Algebra:

**Useful analogy to calculus...**

• You can represent complex functions like **exp(x)** using simpler functions

• If you only get to use **1,x,x2,x3,x4,...** as the pieces...

• ...turns out **exp(x) = 1 + x + x2/2! + x3/3! + …**

• In Calculus, we tell you the general formula, the **Taylor series expansion**

• **f(x) = f(0) + f’(0)/1! x + f’’(0)/2! x2 + f’’’(0)/3! x3 + …**

• If you take more math, you might find out several other ways:

• If it’s a periodic function, can use a **Fourier series**

Shannon Expansion

• **Suppose we have a function F(x1,x2, ..., xn)**

• **Define a new function if we set one of the xi=*constant***

• Example: **F(x1, x2, ..., xi=1, ..., xn)**

• Example: **F(x1, x2, ..., xi=0, ..., xn)**

**Shannon Expansion Theorem**

• Given any Boolean function **F(x1, x2, ..., xn)** and pick any **xi** in **F( )’s** inputs

**F( )** can be represented as

• Pretty easy to prove...

**F(x1, x2, …, xi, …..., xn) = xi • F(xi=1) + xi’ • F(xi=0)**

**BTW, there is notation for these as well**

• Shannon Cofactor with respect to **xi** and **xj**

• Write **F(x1, x2, ..., xi=1, ..., xj=0, ..., xn)** as **Fxi xj’** or **Fxi xj**

• Ditto for any number of variables **xi, xj, xk**, ...

• Notice also that order does **not** matter: **(Fx)y = (Fy)x = Fxy**

• For our example

• Again, **remember**: each of the cofactors is a ***function***, not a number

**F(x,y,z,w) = xy • Fxy + x’y • Fx’y + xy’ • Fxy’ + x’y’ • Fx’y’**

**Fxy = F(x=1, y=1, z, w) = a Boolean *function* of z and w**