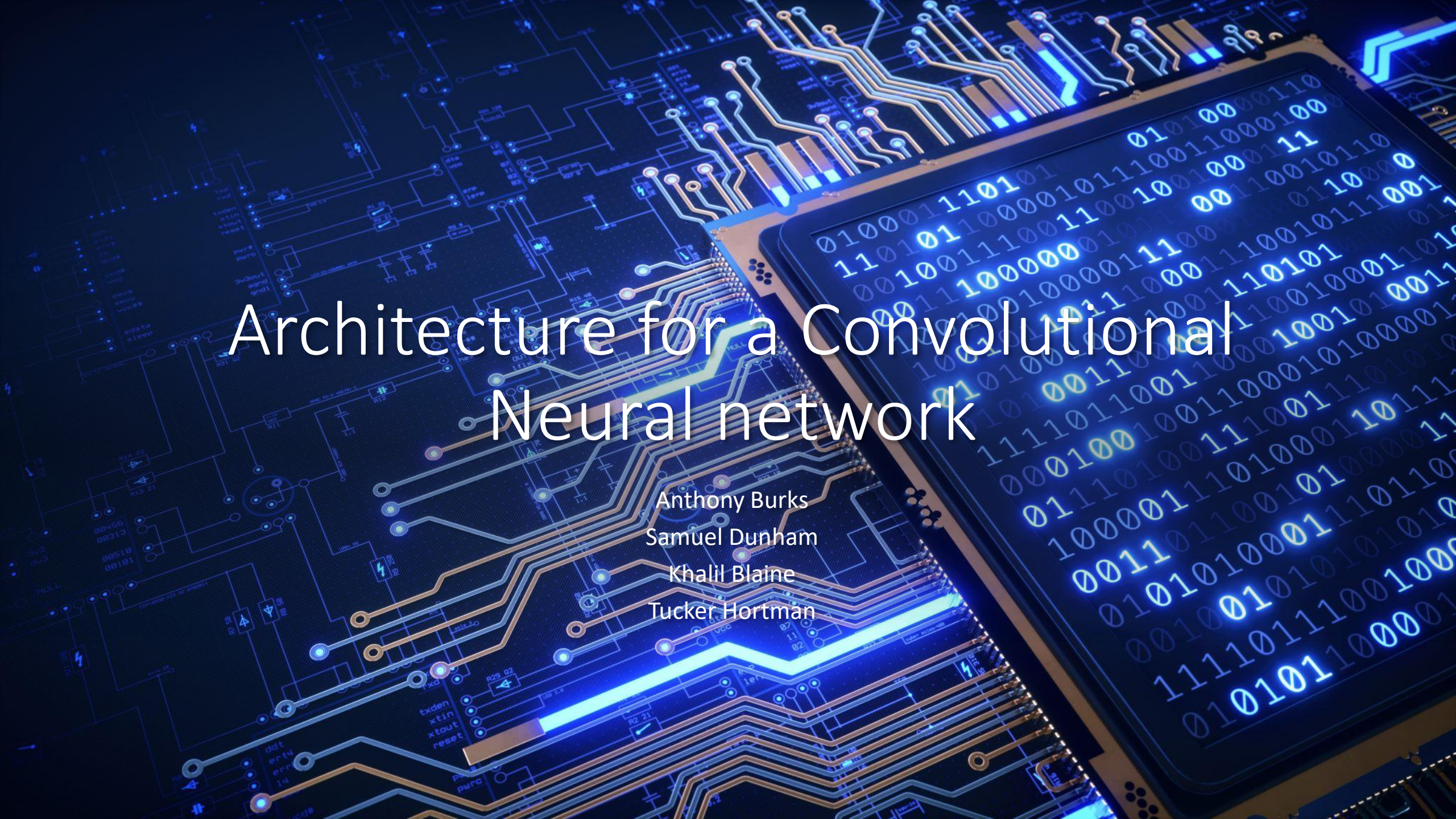


Architecture for a Convolutional Neural network

Anthony Burks
Samuel Dunham
Khalil Blaine
Tucker Hortman

What is Convolution

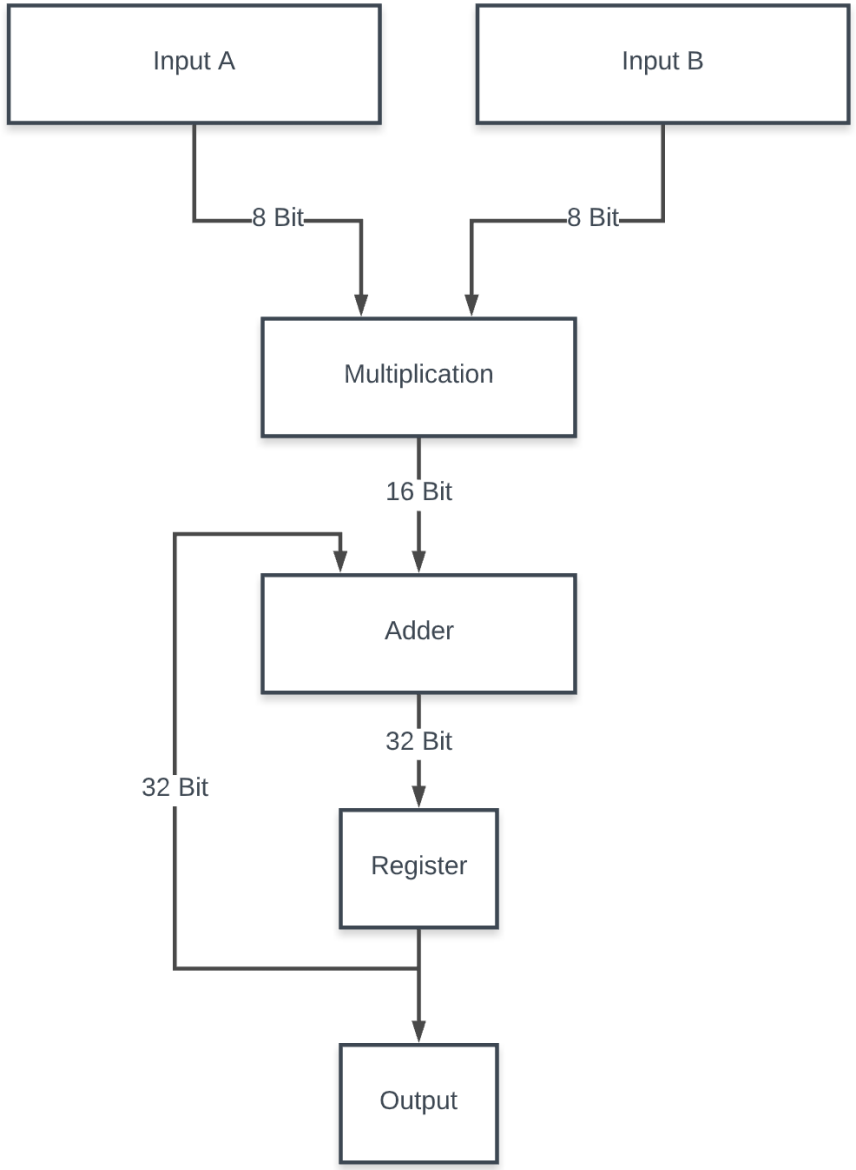
- Convolution is the dot product of two matrixes
- Matrix X convolved with matrix Y is:
  - $X_{00} * Y_{00} + X_{10} * Y_{10} + X_{20} * Y_{20} + X_{01} * Y_{01} + X_{11} * Y_{11} + X_{21} * Y_{21} + X_{02} * Y_{02} + X_{12} * Y_{12} + X_{22} * Y_{22}$

X_{00}	X_{10}	X_{20}
X_{01}	X_{11}	X_{21}
X_{02}	X_{12}	X_{22}

Y_{00}	Y_{10}	Y_{20}
Y_{01}	Y_{11}	Y_{21}
Y_{02}	Y_{12}	Y_{22}

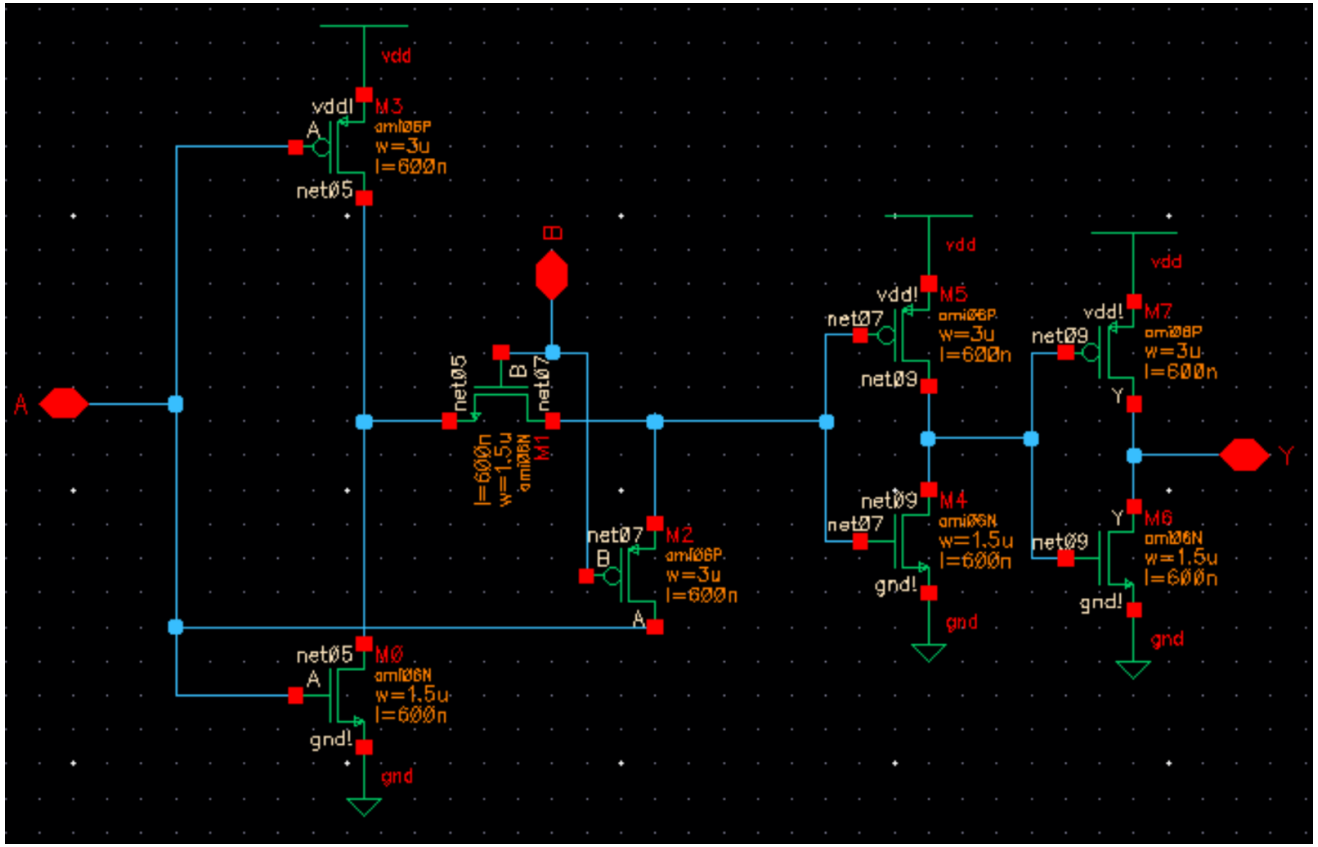
The Plan

- Input A is a cell from matrix X
- Input B is a cell form matrix Y
- They are multiplied together
- The result of the multiplication is then added to the running sum
- The register is then updated with the new sum



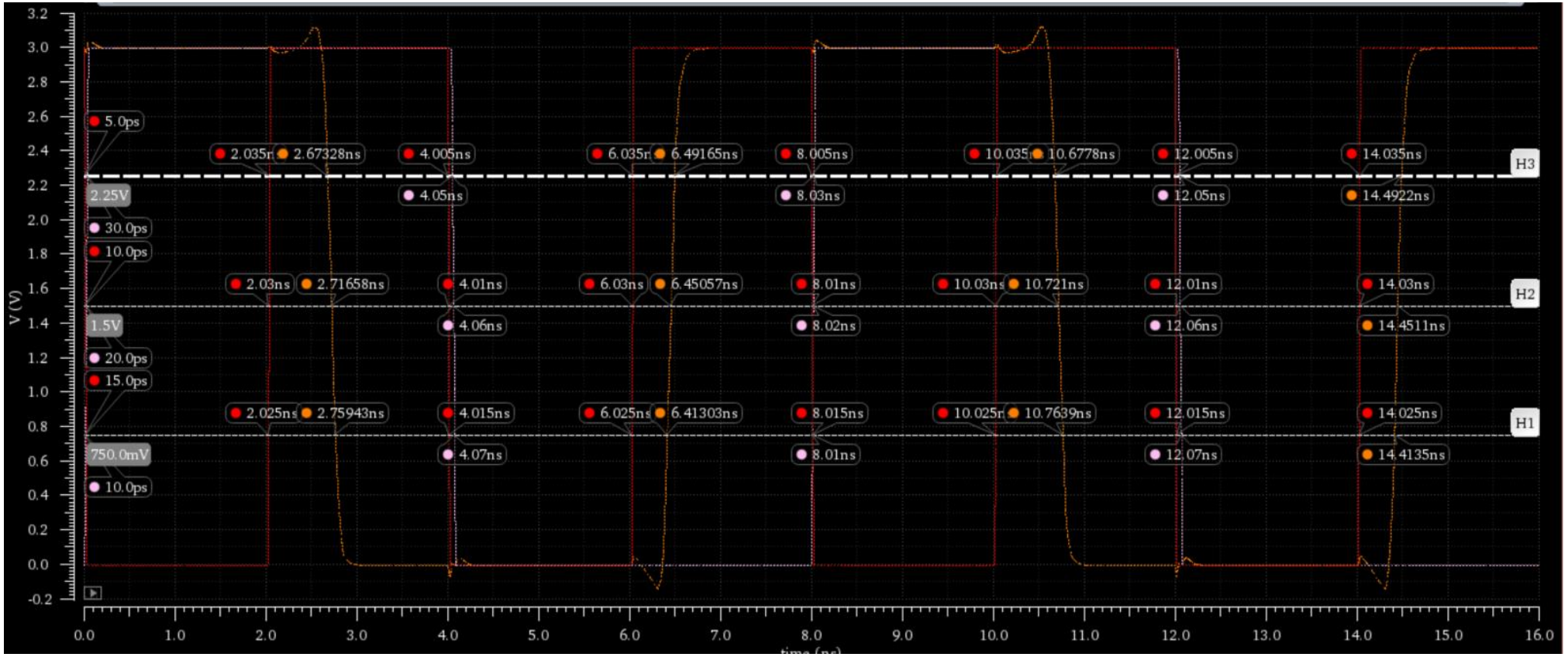
XOR GATE

Used to make the half adder



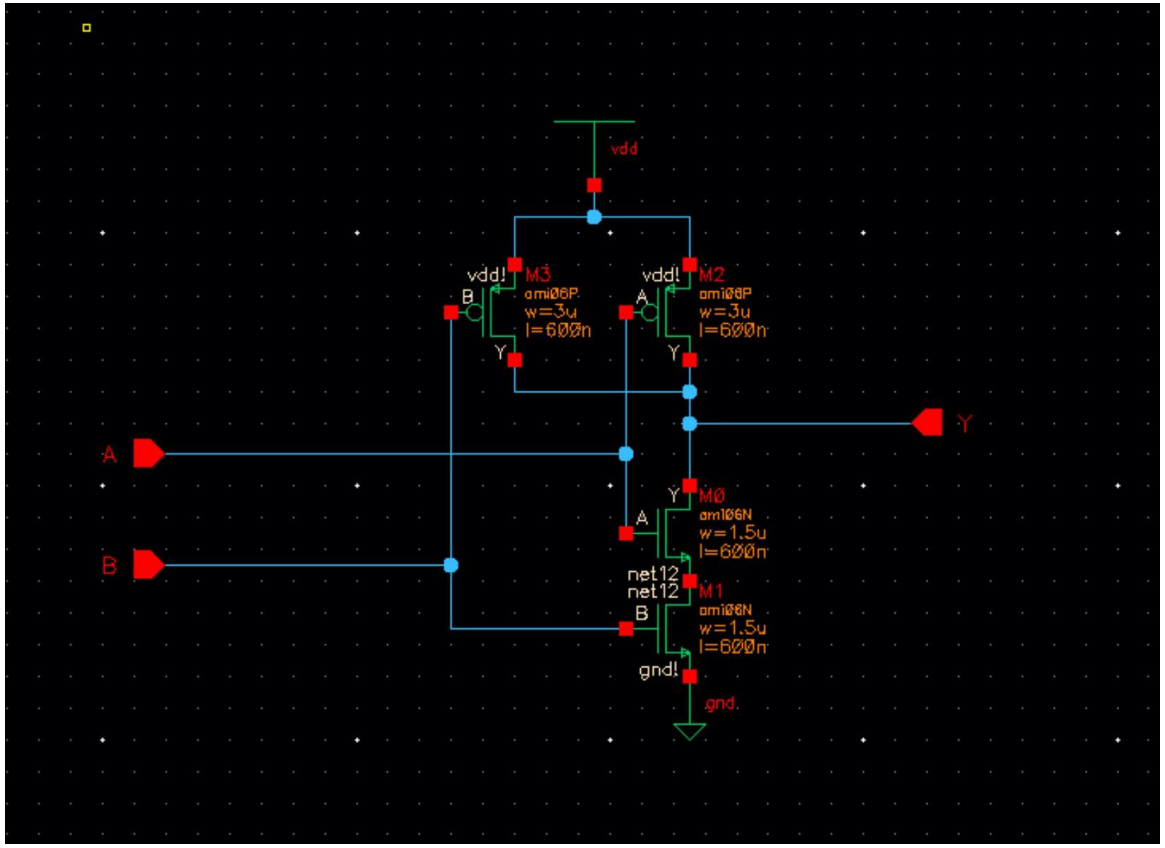
XOR Gate Simulation

Delay – 0.7ns

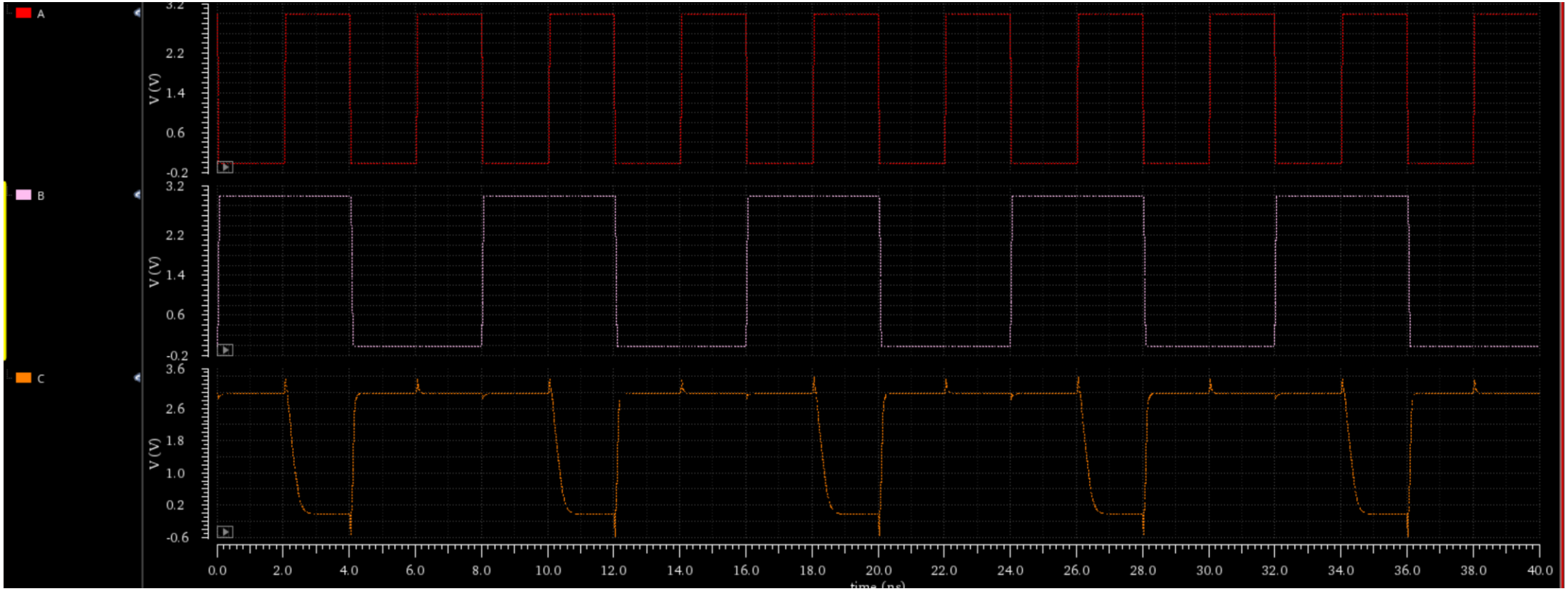


NAND Gate

Used to make the flip-flop and half adder.

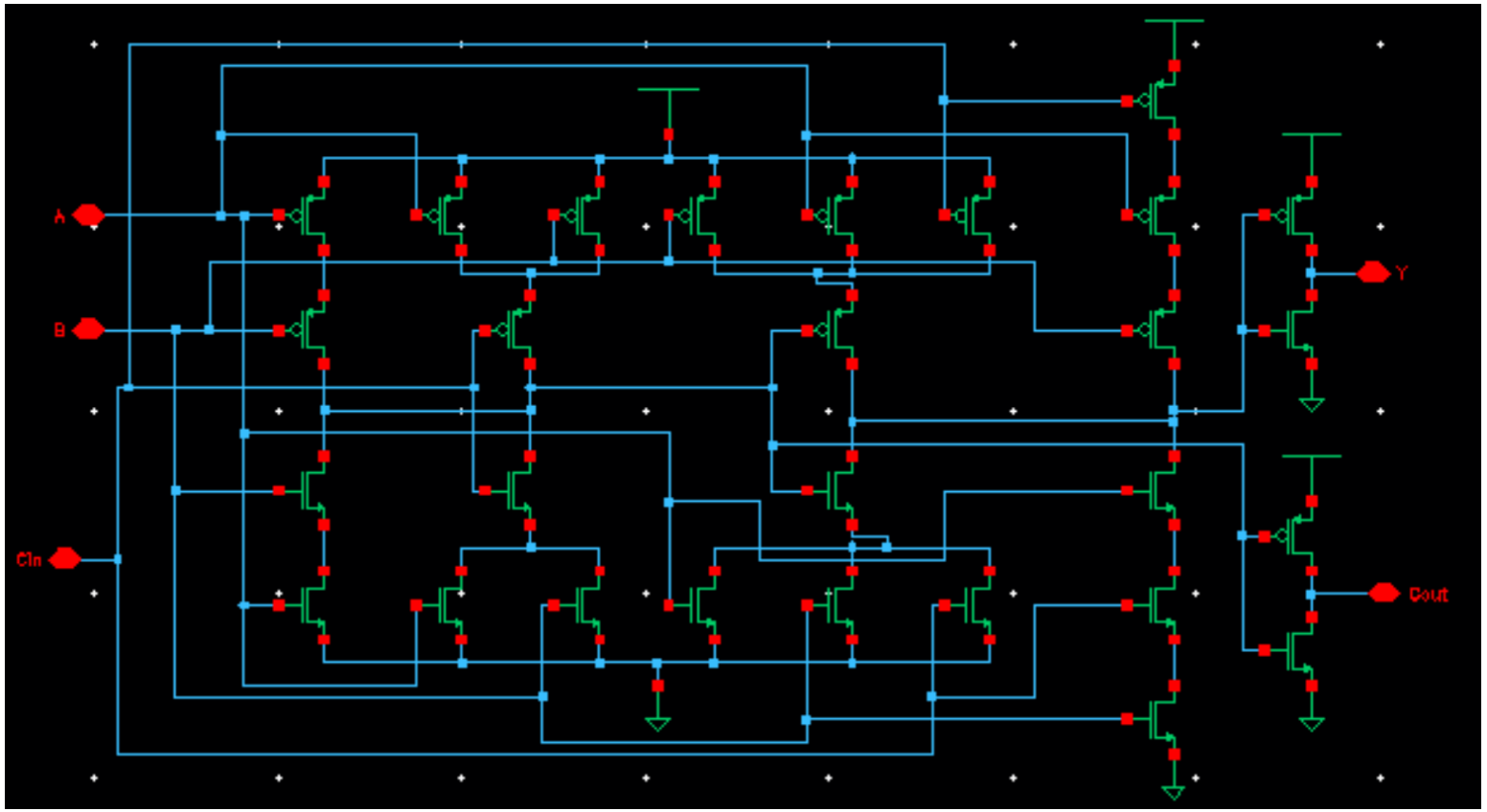


NAND Gate Simulation



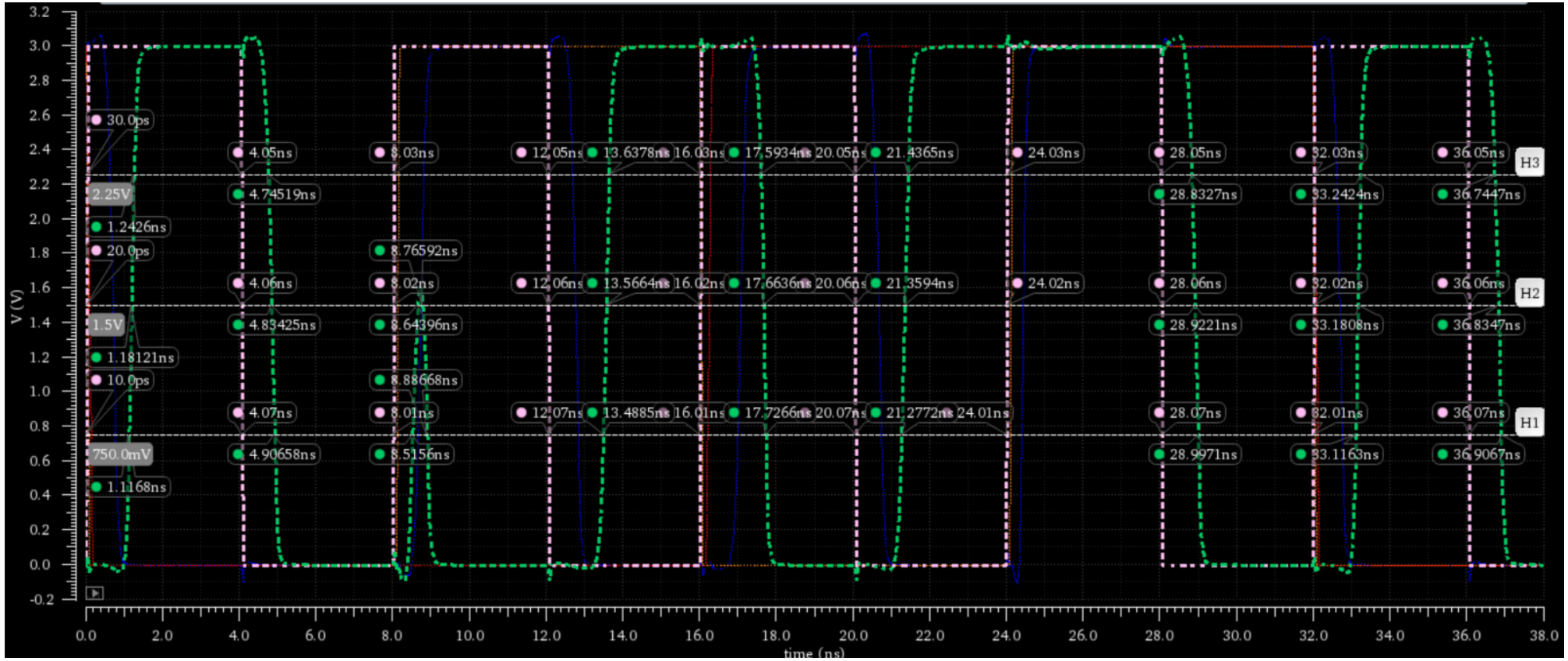
Adder

Used in the multiplier and 32bit adder



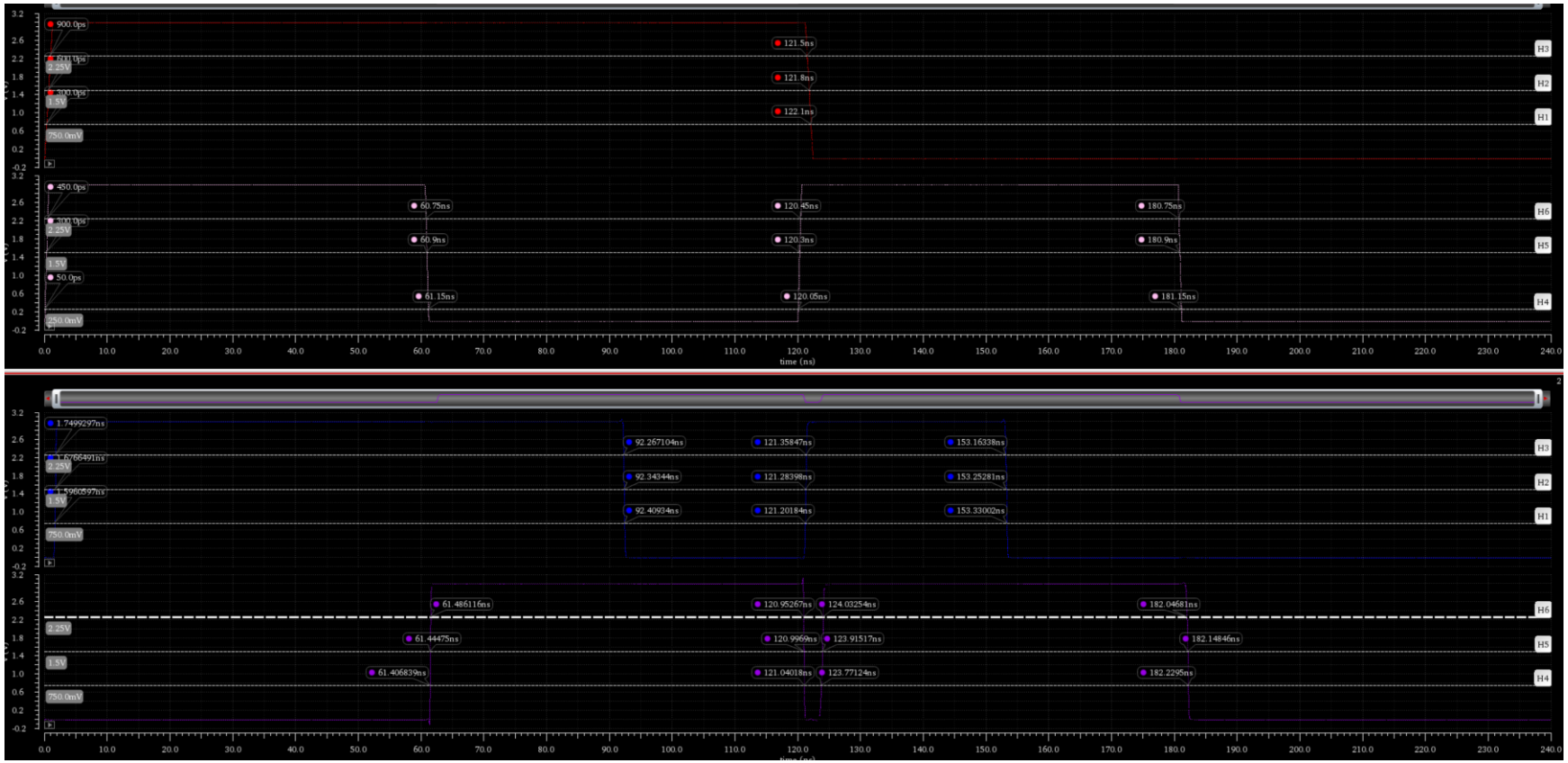
Adder Simulation

Delay – 1.5ns



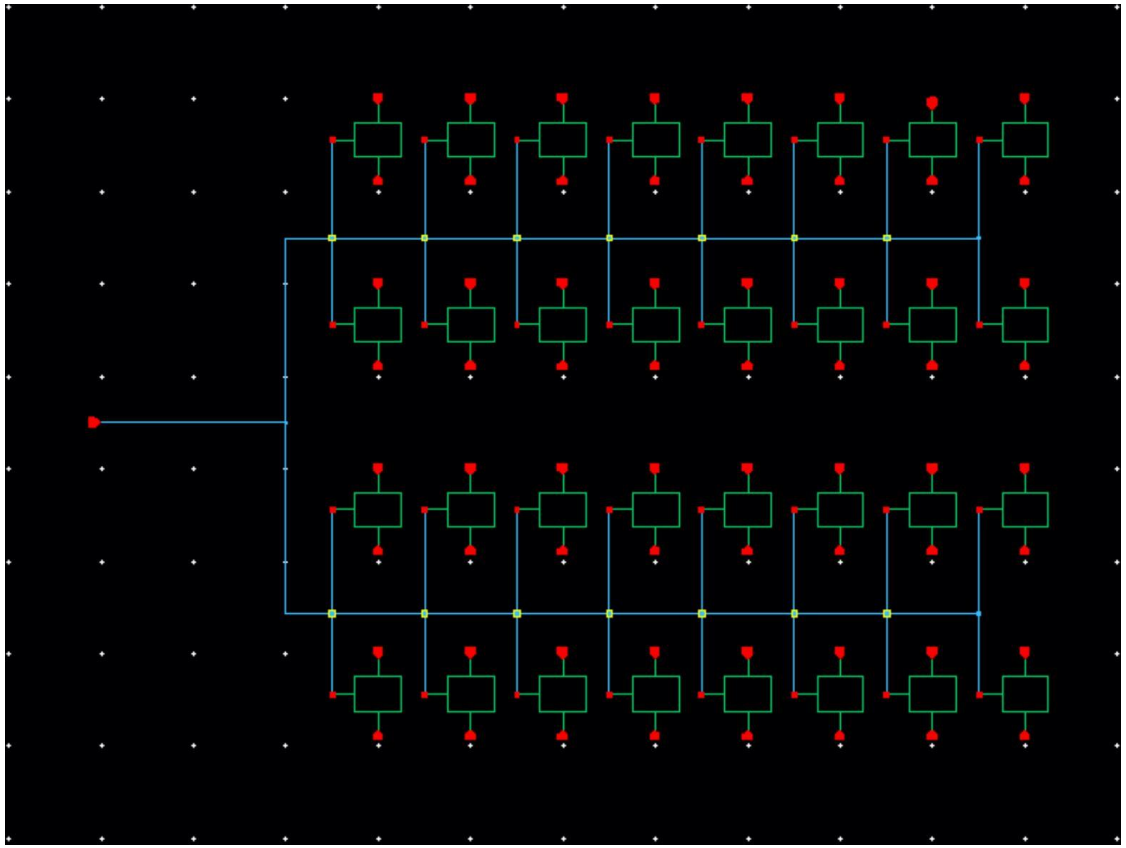
32-bit Adder timing

Delay – 33ns



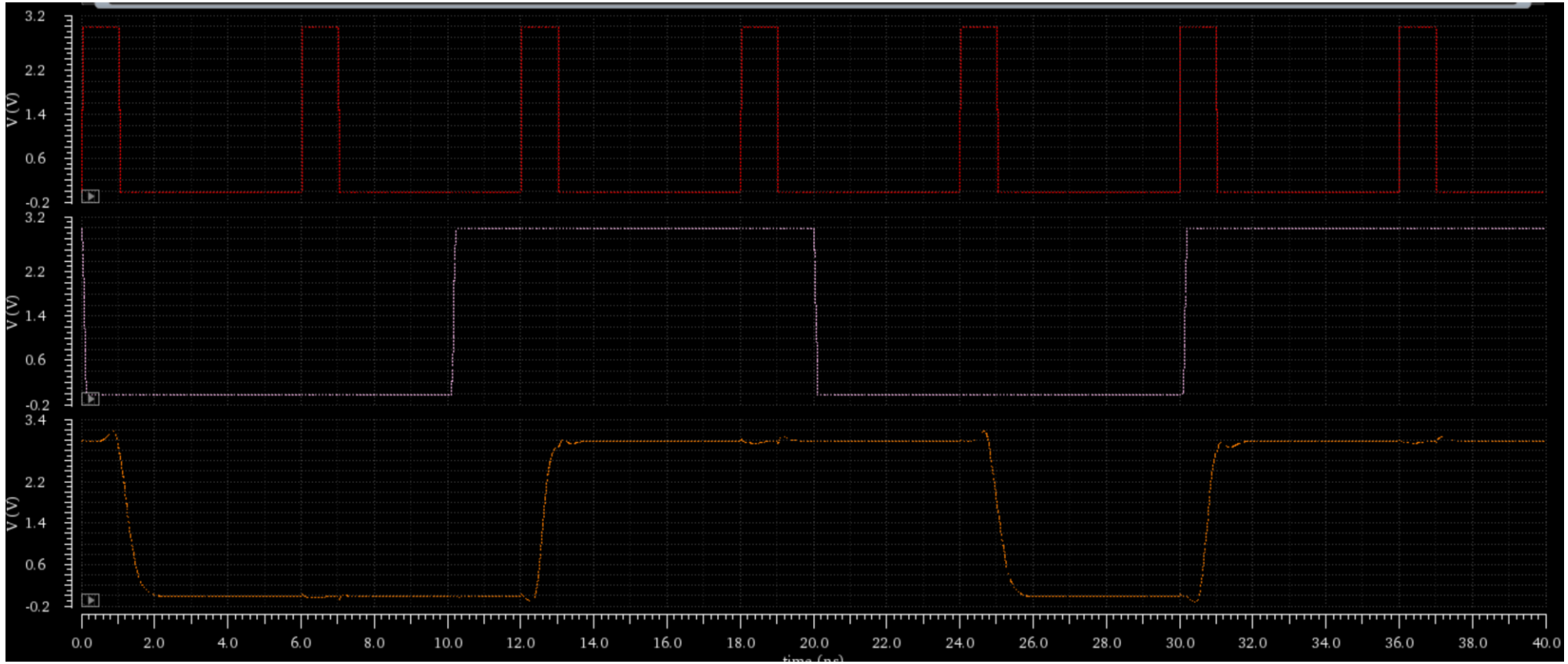
Flip-Flop

Used as a register



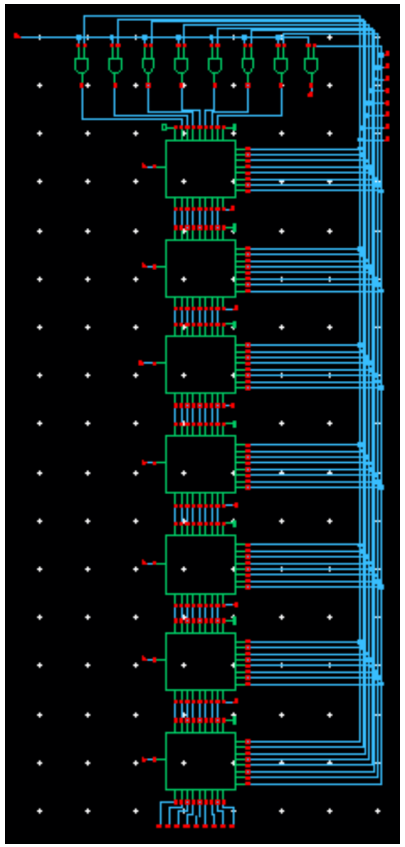
Flip-Flop Simulation

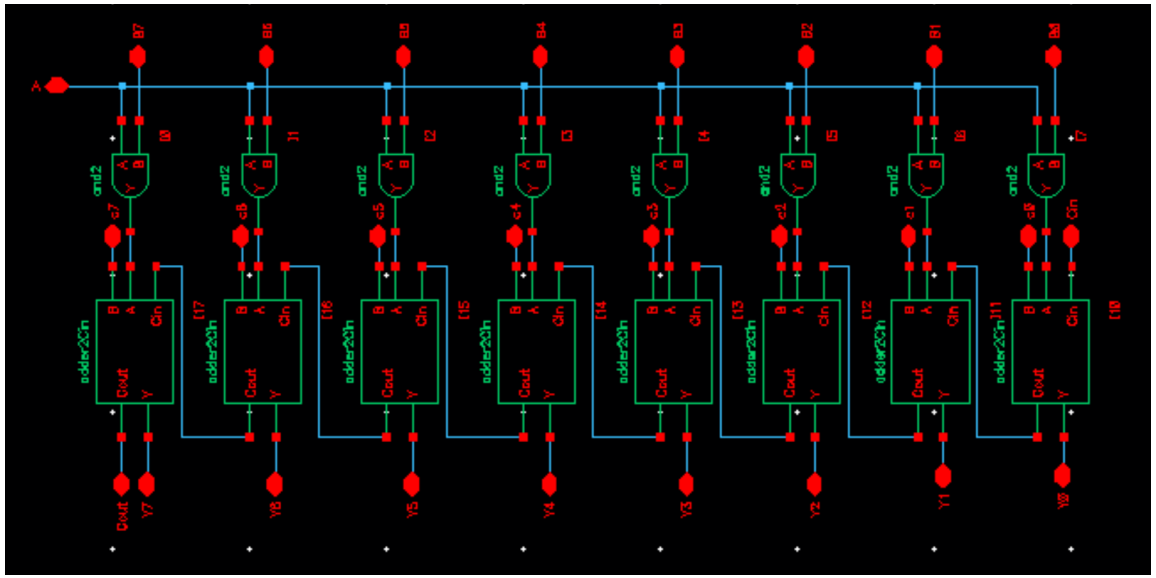
Delay -



Multiplier

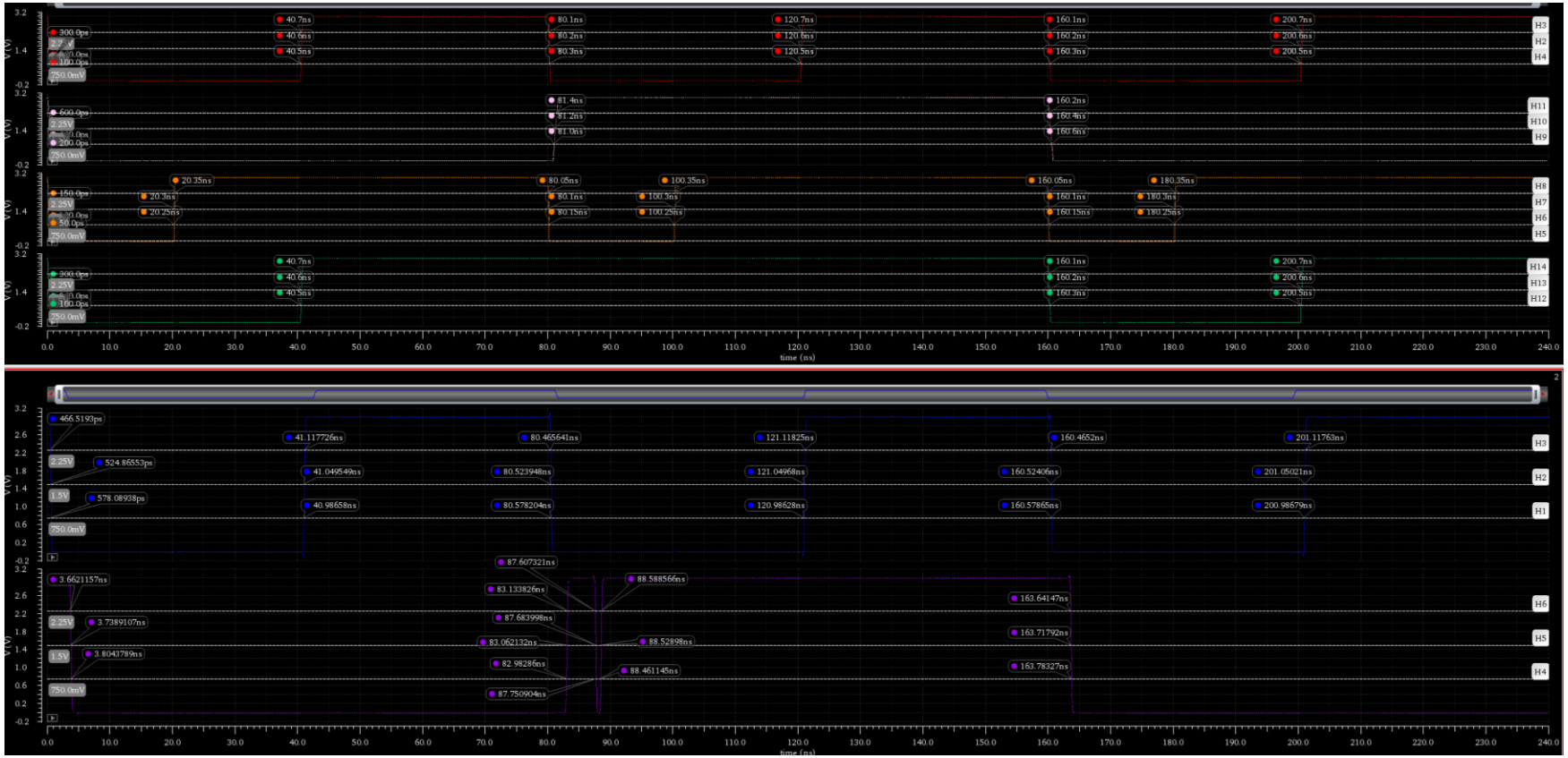
8-bit multiplier



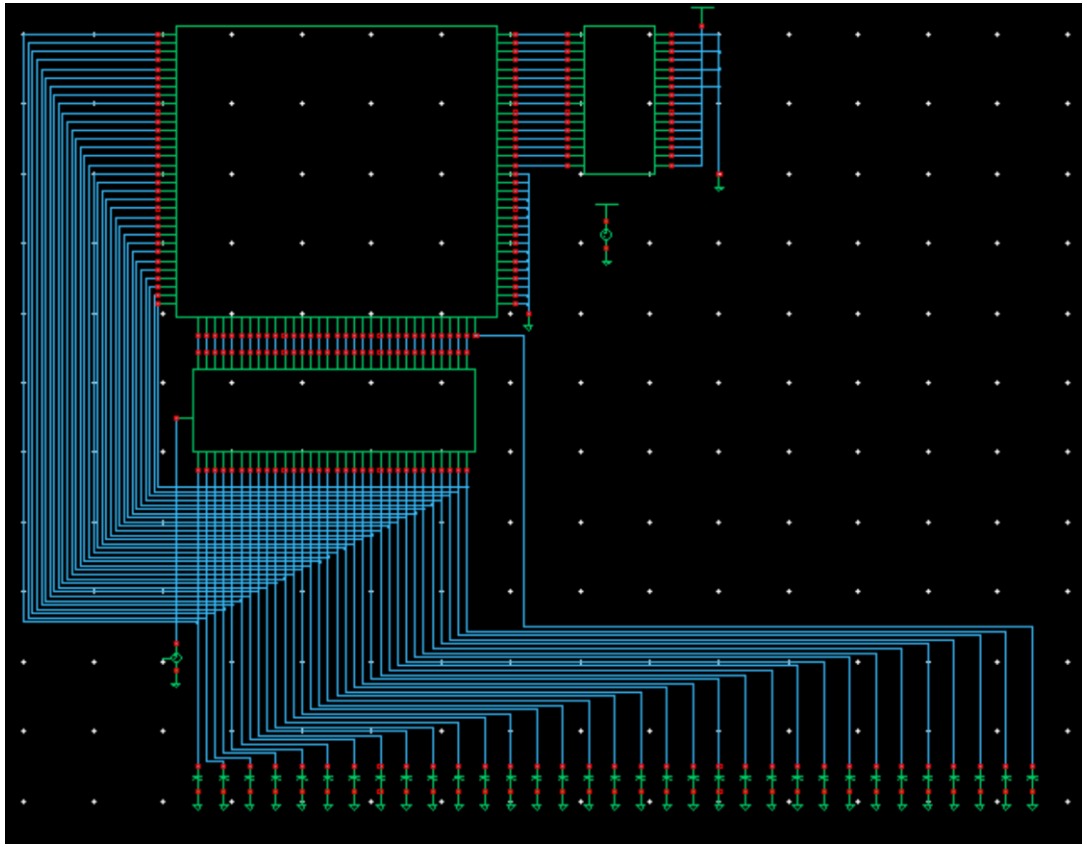


Multiplier Simulation

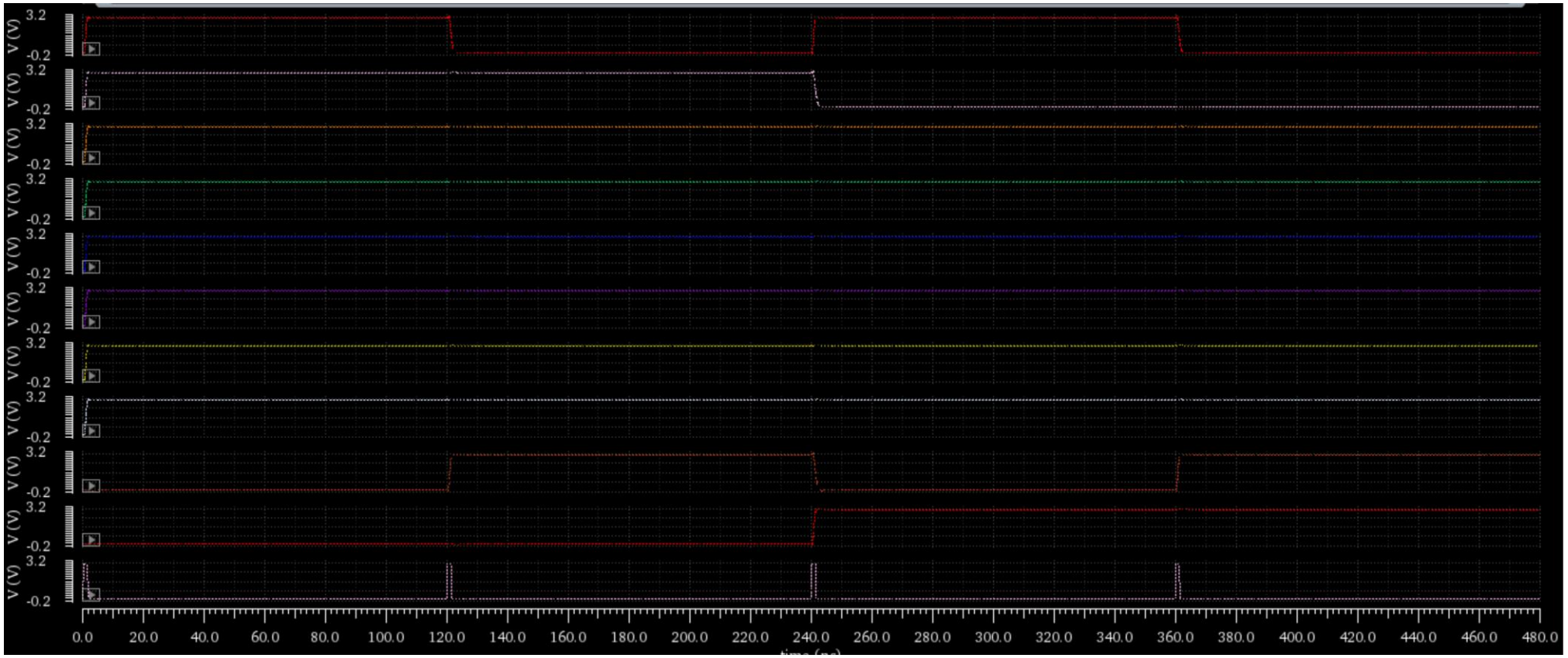
Delay – 11ns



Final Test Bench



The Full System Simulated



Conclusion

- Full system works
- Multiplier and adder need optimization to reduce delay times
- Use of a register to track running sum allows for dynamic kernel size
- Upper limit of kernel size dependent on register size due to overflow