Machine Learning for Error Verification and Correction

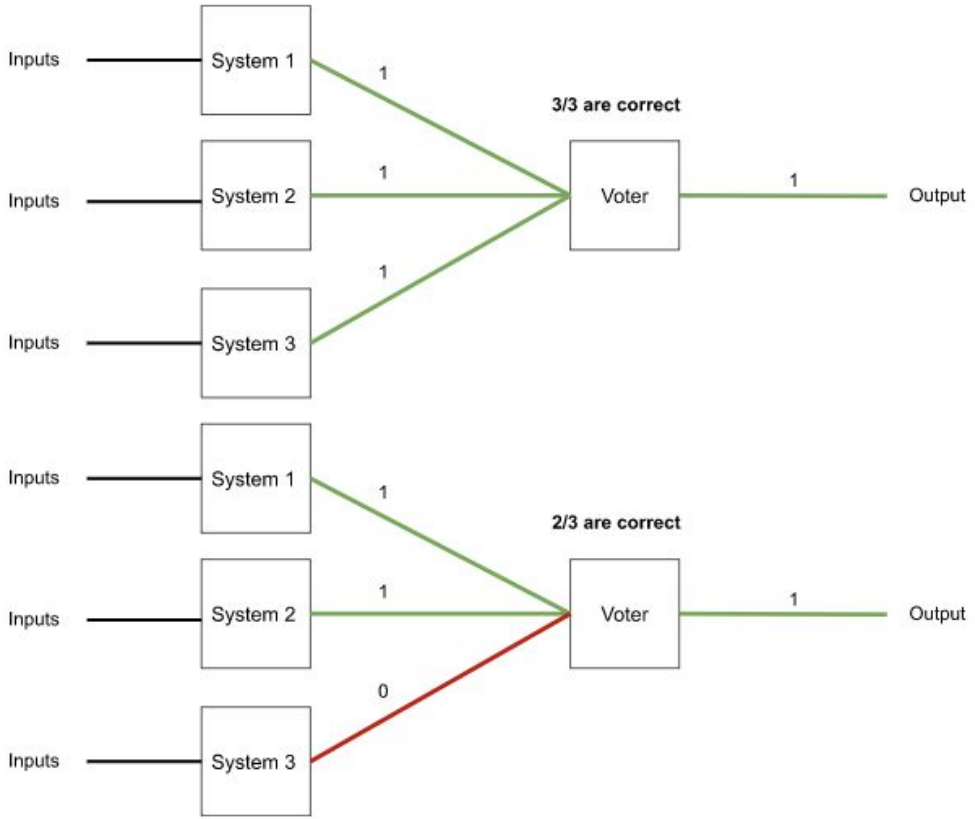
By: Omri Steinberg-Tatman and Jackson Vaughn

Objective

- Using Machine Learning
  - Verification
  - Correction
- Application in circuits
  - Combinational
  - Sequential

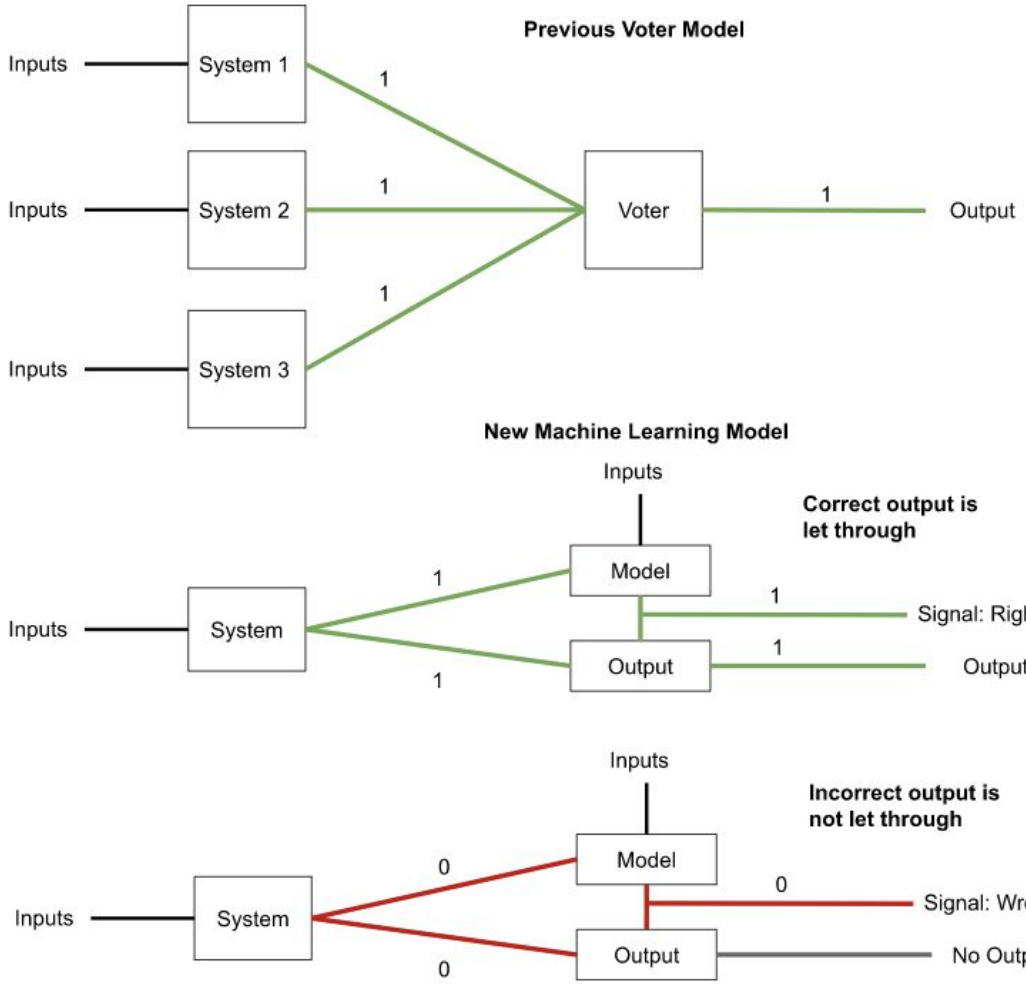
Hardware Voting Systems

- Decides on output
  - Takes in multiple inputs
  - Outputs majority vote
- Inefficient
  - Construction time
  - Construction cost
  - Volume
  - Power draw



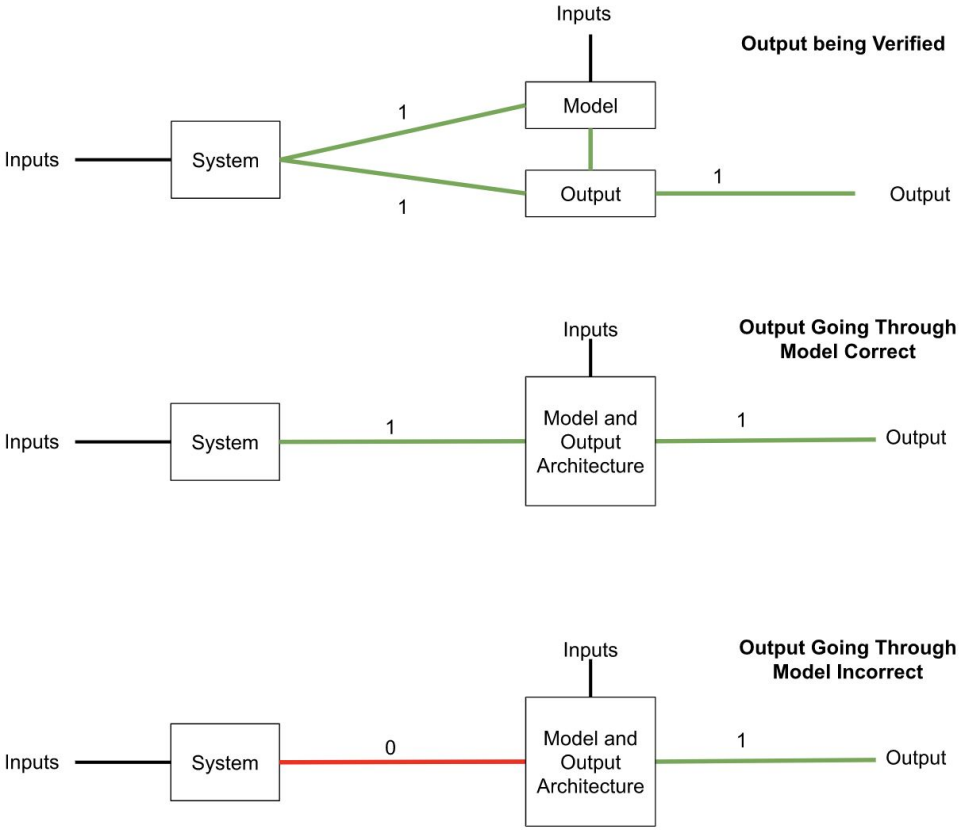
Hardware Verification

- Aim only correct output
  - Two outputs
    - Signal right or wrong
    - Output
- Output will not proceed if wrong
  - Propagation



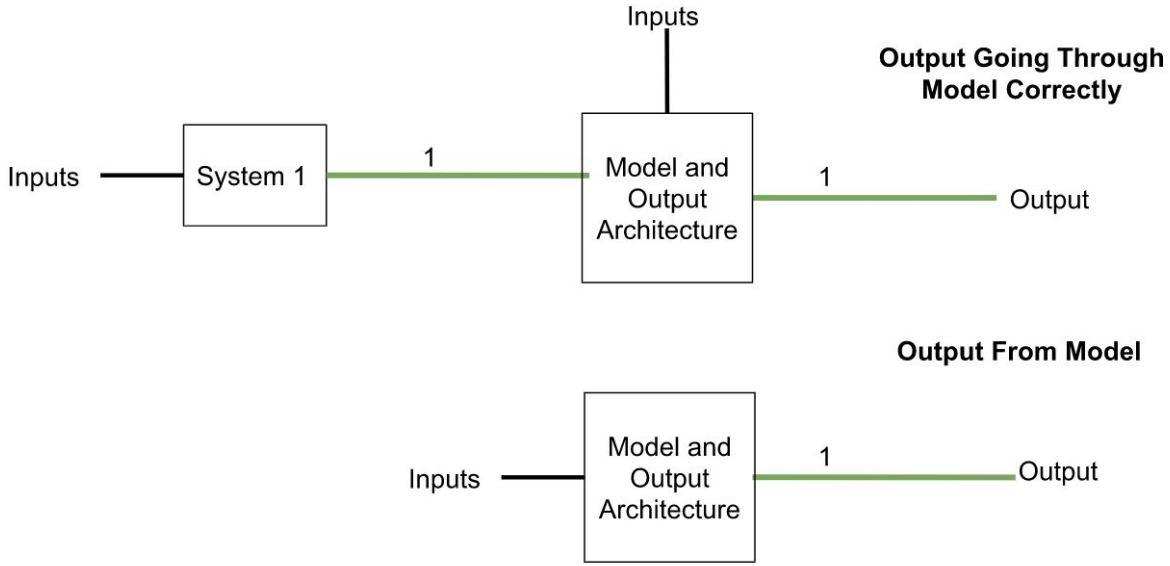
Inefficiencies in Verification Design

- At some points may have no output
  - Can cause issues
- Run output architecture through model
  - Output is known to the model and can always be right



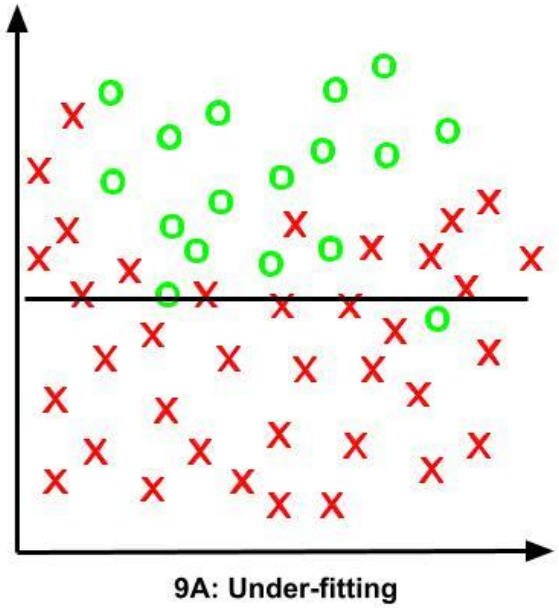
Hardware Correction

- Model has two inputs
  - Original inputs
  - Hardware output
- Bypass the hardware
  - Have model directly pass output

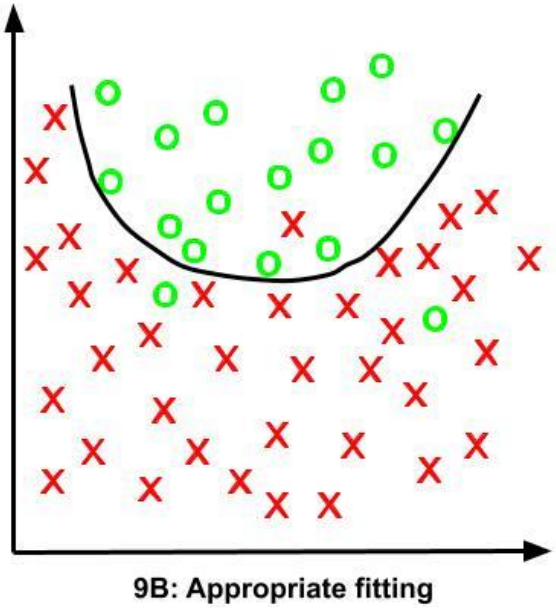


Background

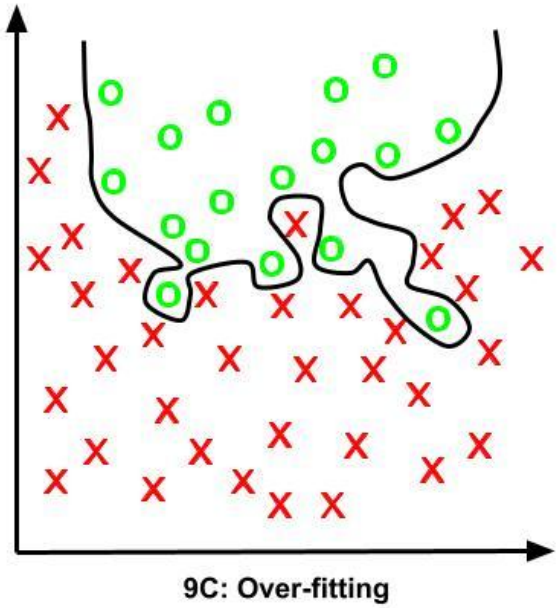
Fitting In The Model



9A: Under-fitting



9B: Appropriate fitting

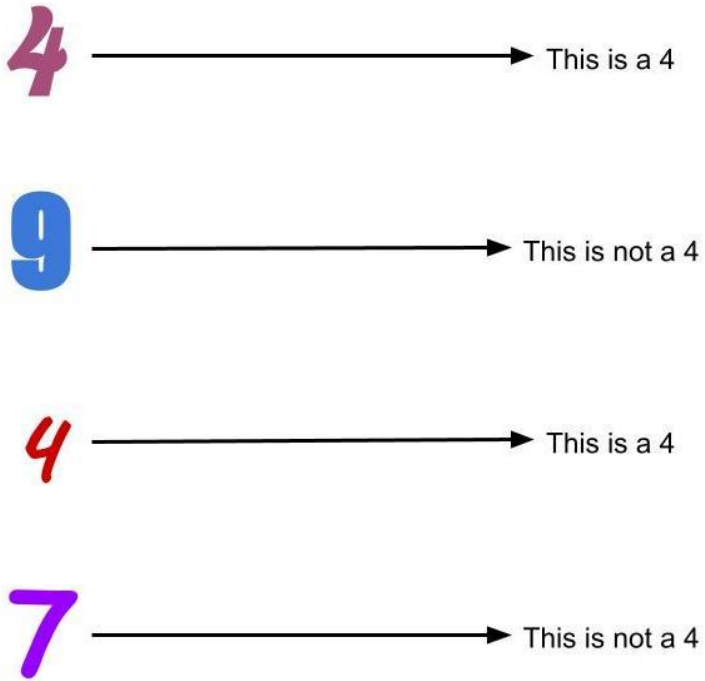


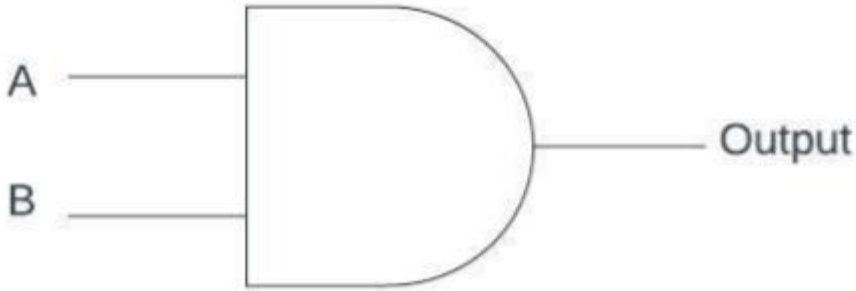
9C: Over-fitting

- Model needs to incorporate intricacies of the data
  - Cannot become too focused on a single dataset

Overfitting

Inputs Outputs





A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Loss vs Accuracy

- Accuracy is fraction of correct guesses
- Loss is difference between perfect certainty and the model's certainty
  - {0.55,0.45} = {1.0,0.0}
  - {0.98,0.02} = {1.0,0.0}
- Accuracy can be perfect while loss is not perfect
  - Too low of a loss can lead to overfitting

Inputs	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Final Test
	{0.55,0.45}	{0.63,0.37}	{0.78,0.22}	{0.90,0.10}	{0.98,0.02}	This is a 4
	{0.55,0.45}	{0.40,0.60}	{0.37,0.63}	{0.22,0.78}	{0.01,0.99}	This is not 4
	{0.55,0.45}	{0.64,0.36}	{0.80,0.20}	{0.88,0.12}	{0.97,0.03}	This is a 4
	{0.55,0.45}	{0.41,0.59}	{0.16,0.84}	{0.11,0.89}	{0.06,0.94}	This is not 4

Machine Learning Application In this Research

- Overfitting works well
  - All possible inputs can be encompassed in a single dataset
- Has a consistent power draw
  - Circuits linearly increase in power consumption
  - Main cost input is for the initial model training
- Model can be remotely updated
  - Model is only a set of weights and biases, can be retrained remotely and the new values sent to be used
- Machine learning hardware is getting better
  - Cheaper
  - Smaller
  - More power efficient

Data Generation

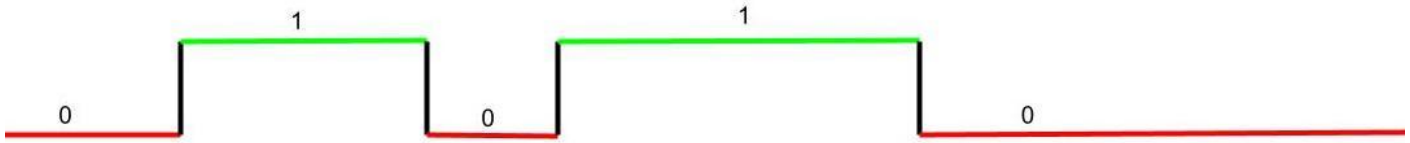
- Use Verilog to generate various modules
- Use testbenches to generate the output for every possible input combination
- Testbenches created CSV of approximately 100,000 lines

Injecting Errors

- Randomly generated outputs for certain inputs.
  - Care had to be taken to ensure that the generated output was not the same as the original output.

Proof Of Concept

HighLow Model

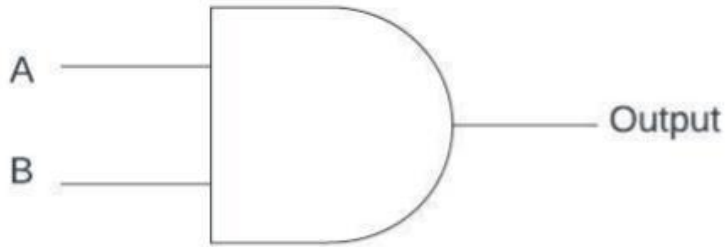



```
Class 0:
  Total: 4904
  Correct: 4904
  True 0: 4904
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 5095
  Correct: 5095
  True 1: 5095
  False 0: 0
  Accuracy: 1.0
```

```
loss: 6.6736e-04 - accuracy: 1.0000
```

Combinational

AND Gate Model



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

```
Class 0:
  Total: 4954
  Correct: 4954
  True 0: 4954
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 5045
  Correct: 5045
  True 1: 5045
  False 0: 0
  Accuracy: 1.0

loss: 1.0585e-06 - accuracy: 1.0000
```

Verification Model

```
Class 0:
  Total: 7477
  Correct: 7477
  True 0: 7477
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 2522
  Correct: 2522
  True 1: 2522
  False 0: 0
  Accuracy: 1.0

loss: 1.6351e-04 - accuracy: 1.0000
```

Correction Model

-> Roughly 3 Epochs to attain 100% accuracy

OR Gate Model

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

```
Class 0:
  Total: 5048
  Correct: 5048
  True 0: 5048
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 4951
  Correct: 4951
  True 1: 4951
  False 0: 0
  Accuracy: 1.0
loss: 1.6073e-04 - accuracy: 1.0000
```

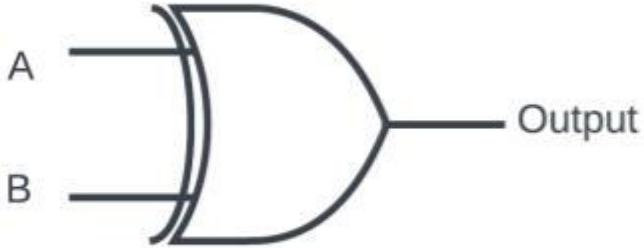
Verification Model

```
Class 0:
  Total: 2518
  Correct: 2518
  True 0: 2518
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 7481
  Correct: 7481
  True 1: 7481
  False 0: 0
  Accuracy: 1.0
loss: 1.6268e-04 - accuracy: 1.0000
```

Correction Model

-> Roughly 3 Epochs to attain 100% accuracy

XOR Gate Model



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

```
Class 0:
  Total: 4933
  Correct: 4933
  True 0: 4933
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 5066
  Correct: 5066
  True 1: 5066
  False 0: 0
  Accuracy: 1.0

loss: 1.2419e-06 - accuracy: 1.0000
```

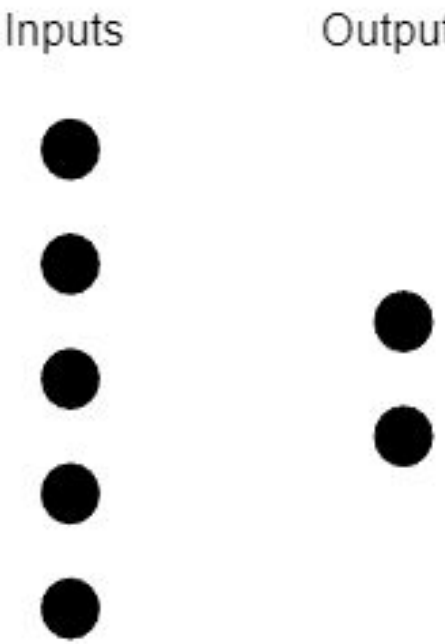
Verification Model

```
Class 0:
  Total: 4918
  Correct: 4918
  True 0: 4918
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 5081
  Correct: 5081
  True 1: 5081
  False 0: 0
  Accuracy: 1.0

loss: 1.0376e-06 - accuracy: 1.0000
```

Correction Model

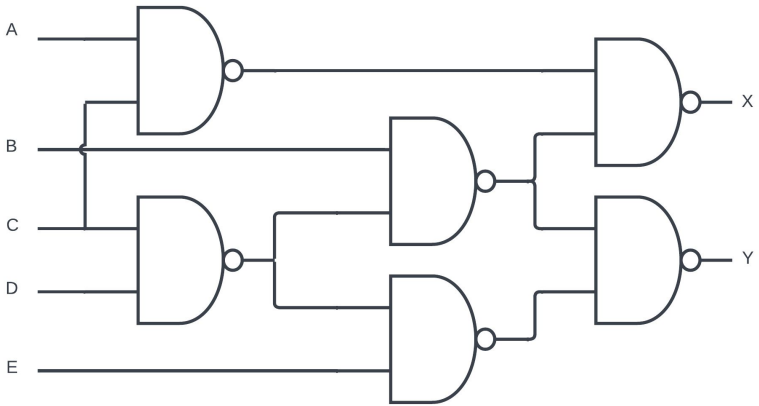
-> Roughly 3 Epochs to attain 100% accuracy

Classes



Possible Output States	Mapped Classes
0,0	0
0,1	1
1,0	2
1,1	3

C17 Model




```
Class 0:
  Total: 11583
  Correct: 11583
  True 0: 11583
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 11748
  Correct: 11748
  True 1: 11748
  False 0: 0
  Accuracy: 1.0
loss: 8.8454e-07 - accuracy: 1.0000
```

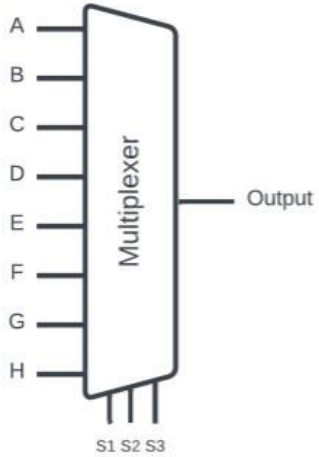
Verification Model

```
Class 0:
  Total: 6571
  Correct: 6571
  True 0: 6571
  False 1: 0
  False 2: 0
  False 3: 0
  Accuracy: 1.0
Class 1:
  Total: 3602
  Correct: 3602
  True 1: 3602
  False 0: 0
  False 2: 0
  False 3: 0
  Accuracy: 1.0
Class 2:
  Total: 3597
  Correct: 3597
  True 2: 3597
  False 0: 0
  False 1: 0
  False 3: 0
  Accuracy: 1.0
Class 3:
  Total: 9561
  Correct: 9561
  True 3: 9561
  False 0: 0
  False 1: 0
  False 2: 0
  Accuracy: 1.0
loss: 7.1704e-07 - accuracy: 1.0000
```

Correction Model

-> Roughly 5 Epochs to attain 100% accuracy

8-TO-1 Multiplexer Model



S1	S2	S3	Output
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	G
1	1	1	H

```
Class 0:
  Total: 5048
  Correct: 5048
  True 0: 5048
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 4951
  Correct: 4951
  True 1: 4951
  False 0: 0
  Accuracy: 1.0

loss: 1.6073e-04 - accuracy: 1.0000
```

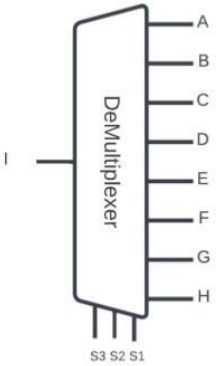
Verification Model

```
Class 0:
  Total: 16718
  Correct: 16718
  True 0: 16718
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 16612
  Correct: 16612
  True 1: 16612
  False 0: 0
  Accuracy: 1.0

loss: 5.7934e-07 - accuracy: 1.0000
```

Correction Model

-> Roughly 10 Epochs to attain 100% accuracy

1-TO-8 DeMultiplexer Model



I	S1	S2	S3	A	B	C	D	E	F	G	H
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1
0	X	X	X	0	0	0	0	0	0	0	0

```
Class 0:
  Total: 25055
  Correct: 25055
  True 0: 25055
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 24945
  Correct: 24945
  True 1: 24945
  False 0: 0
  Accuracy: 1.0
```

```
loss: 3.1348e-06 - accuracy: 1.0000
```

Verification Model

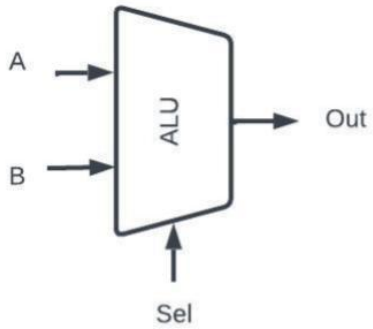
```
Class 0:
  Total: 16613
  Correct: 16613
  True 0: 16613
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 1:
  Total: 2097
  Correct: 2097
  True 1: 2097
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 2:
  Total: 2117
  Correct: 2117
  True 2: 2117
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 3:
  Total: 2048
  Correct: 2048
  True 3: 2048
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 4:
  Total: 2047
  Correct: 2047
  True 4: 2047
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 5:
  Total: 2110
  Correct: 2110
  True 5: 2110
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 6:
  Total: 2112
  Correct: 2112
  True 6: 2112
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 7:
  Total: 2079
  Correct: 2079
  True 7: 2079
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
Class 8:
  Total: 2107
  Correct: 2107
  True 8: 2107
  False: [0, 0, 0, 0, 0, 0, 0, 0, 0]
  Accuracy: 1.0
```

```
loss: 4.6966e-07 - accuracy: 1.0000
```

Correction Model

-> Roughly 10 Epochs to attain 100% accuracy

ALU Model



Sel	Out
0000	A + B
0001	A - B
0010	A * B
0011	A / B
0100	A << 1
0101	A >> 1
0110	A rot left by 1
0111	A rot right by 1

Sel	Out
1000	A and B
1001	A or B
1010	A xor B
1011	A nor B
1100	A nand B
1101	A xnor B
1110	Out = 1 if A > B
1111	Out = 1 if A = B

```
Class 0:
  Total: 82028
  Correct: 82028
  True 0: 82028
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 81812
  Correct: 81812
  True 1: 81812
  False 0: 0
  Accuracy: 1.0

loss: 4.7324e-06 - accuracy: 1.0000
```

Verification Model

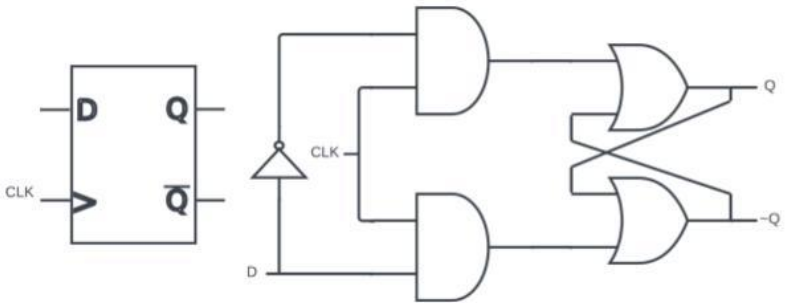
```
Class 0:
  Accuracy: 1.0
Class 1:
  Accuracy: 1.0
Class 2:
  Accuracy: 1.0
Class 3:
  Accuracy: 1.0
Class 4:
  Accuracy: 1.0
Class 5:
  Accuracy: 1.0
Class 6:
  Accuracy: 1.0
Class 7:
  Accuracy: 1.0
Class 8:
  Accuracy: 1.0
Class 9:
  Accuracy: 1.0
Class 10:
  Accuracy: 1.0
Class 11:
  Accuracy: 1.0
Class 12:
  Accuracy: 1.0
Class 13:
  Accuracy: 1.0
Class 14:
  Accuracy: 1.0
Class 15:
  Accuracy: 1.0
Class 16:
  Accuracy: 1.0
Class 17:
  Accuracy: 1.0
Class 18:
  Accuracy: 1.0
Class 19:
  Accuracy: 1.0
Class 20:
  Accuracy: 1.0
Class 21:
  Accuracy: 1.0
Class 22:
  Accuracy: 1.0
Class 23:
  Accuracy: 1.0
Class 24:
  Accuracy: 1.0
Class 25:
  Accuracy: 1.0
Class 26:
  Accuracy: 1.0
Class 27:
  Accuracy: 1.0
Class 28:
  Accuracy: 1.0
Class 29:
  Accuracy: 1.0
Class 30:
  Accuracy: 1.0
Class 31:
  Accuracy: 1.0
Class 32:
  Accuracy: 1.0
Class 33:
  Accuracy: 1.0
Class 34:
  Accuracy: 1.0
Class 35:
  Accuracy: 1.0
Class 36:
  Accuracy: 1.0
Class 37:
  Accuracy: 1.0
Class 38:
  Accuracy: 1.0
Class 39:
  Accuracy: 1.0
Class 40:
  Accuracy: 1.0
Class 41:
  Accuracy: 1.0
Class 42:
  Accuracy: 1.0
Class 43:
  Accuracy: 1.0
Class 44:
  Accuracy: 1.0
Class 45:
  Accuracy: 1.0
Class 46:
  Accuracy: 1.0
Class 47:
  Accuracy: 1.0
Class 48:
  Accuracy: 1.0
Class 49:
  Accuracy: 1.0
Class 50:
  Accuracy: 1.0
Class 51:
  Accuracy: 1.0
Class 52:
  Accuracy: 1.0
Class 53:
  Accuracy: 1.0
Class 54:
  Accuracy: 1.0
Class 55:
  Accuracy: 1.0
Class 56:
  Accuracy: 1.0
Class 57:
  Accuracy: 1.0
Class 58:
  Accuracy: 1.0
Class 59:
  Accuracy: 1.0
Class 60:
  Accuracy: 1.0
Class 61:
  Accuracy: 1.0
Class 62:
  Accuracy: 1.0
Class 63:
  Accuracy: 1.0
Class 64:
  Accuracy: 1.0

loss: 8.2208e-04 - accuracy: 1.0000
```

Correction Model

-> Roughly 1000 Epochs to attain 100% accuracy

Sequential

DFF Model



CLK	D	Q	~Q
↓	X	Q	~Q
↑	0	0	1
↑	1	1	0

```
Class 0:
  Total: 16638
  Correct: 16638
  True 0: 16638
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 16692
  Correct: 16692
  True 1: 16692
  False 0: 0
  Accuracy: 1.0
```

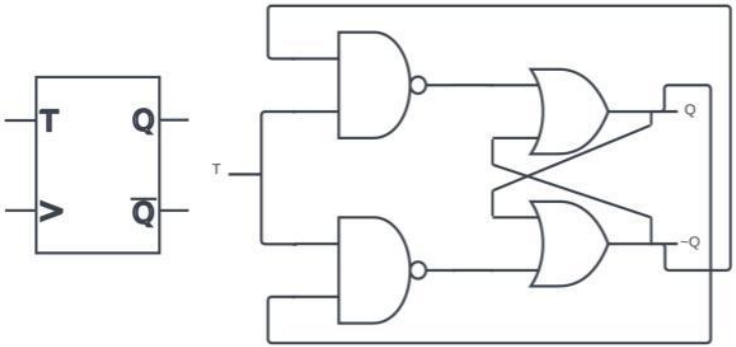
```
Class 0:
  Total: 5559
  Correct: 5559
  True 0: 5559
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 27771
  Correct: 27771
  True 1: 27771
  False 0: 0
  Accuracy: 1.0
```

```
loss: 5.2936e-06 - accuracy: 1.0000 loss: 1.0865e-06 - accuracy: 1.0000
```

Verification Model **Correction Model**

-> Roughly 3 Epochs to attain 100% accuracy

TFF Model



T	Q	Q+
0	0	0
0	1	1
1	0	1
1	1	0

```
Class 0:
  Total: 16648
  Correct: 16648
  True 0: 16648
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 16682
  Correct: 16682
  True 1: 16682
  False 0: 0
  Accuracy: 1.0

loss: 6.0118e-07 - accuracy: 1.0000
```

Verification Model

```
Class 0:
  Total: 16496
  Correct: 16496
  True 0: 16496
  False 1: 0
  Accuracy: 1.0
Class 1:
  Total: 16834
  Correct: 16834
  True 1: 16834
  False 0: 0
  Accuracy: 1.0

loss: 1.0268e-06 - accuracy: 1.0000
```

Correction Model

-> Roughly 3 Epochs to attain 100% accuracy

Conclusion

Outcome

- Models achieved perfect accuracy with little loss in all cases
  - Predictions tested after training were all perfect
- Both combinational and sequential worked
- Large and small circuits performed equally
  - Larger circuits took longer to train

Application In The World

- Replacement of voter systems
  - Theoretically can reduce power consumption at scale
- Replacement of FPGA
  - Emulate with machine learning model instead
- Sensitive hardware systems
  - Saturn V rocket had a voter system with 3 sets of hardware
    - Could instead have 3 machine learning models running

Issues

- Machine learning still has to run on hardware
  - Processor, GPU
- Machine learning does not guarantee accuracy
  - Properly built hardware does
- Initial cost of training is high
  - Data processing
  - Model training
- All modules tested were relatively small and simple
  - Requires much more extensive testing

Further Research

- Partial datasets
  - 25%, 50%, 75%
  - Can the model still emulate general behaviour
    - ALU
- Power consumption
  - Hard built circuit vs FPGA vs Model
- Circuit Emulation
  - FPGA vs Machine Learning Model
    - Speed in setting up
    - Edge cases

Acknowledgments

Professor Hussain Al-Asaad

Gali Steinberg-Tatman

Hans O'Flaherty

Bella Doherty

Aman Ganapathy

Alexander Gardner

Marji K. LeGrand

Questions?