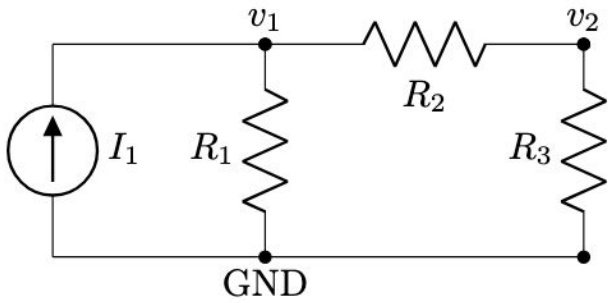
CircuitSim

Andrew Yang
Case Schemmer
Faustina Cheng
Jary Tolentino
Ming Gong

(Modified) Node Voltage Analysis



System of Equations:

$$\begin{cases} G_1 v_1 + G_2(v_1 - v_2) = I_1 \\ G_3 v_2 + G_2(v_2 - v_1) = 0 \end{cases}$$

Kirchhoff's Current Law:

$$\frac{v_1}{R_1} + \frac{v_1 - v_2}{R_2} - I_1 = 0$$

$$\frac{v_2}{R_2} + \frac{v_2 - v_1}{R_3} = 0$$

$$\begin{bmatrix} G_1 + G_2 & -G_2 \\ -G_2 & G_2 + G_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ 0 \end{bmatrix}$$

$$G\mathbf{v} = \mathbf{I}$$

Overall Design

Driver:

1. Takes in input file (SPICE netlist)
    a. Circuit components + transient analysis parameters
2. Translates the components into matrix representation
3. Matrix solver loop
    a. Matrix is written to FPGA memory
    b. FPGA performs Gaussian Elimination
    c. FPGA writes results
4. Writes to CSV file (time, node/voltage values)

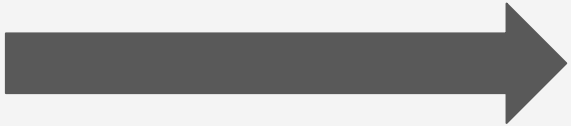
Input Parsing

Input: SPICE deck

```
* Simple LRC

* Components
V1 1 0 DC 5;
R1 1 2 10;
L1 2 3 1e-3;
C1 3 0 1e-6;

.TRAN 5e-6 0.50

* End of Netlist
.END
```

Parse:

```
add_vsrc(...)
add_res(...)
add_ind(...)
add_cap(...)
```



Component Array:

Vsrc
type: Linear
stamp_lin: vsrc_stamp
n1: 1 n2: 0
ni: 4
dc_value: 5

Res
type: Static
stamp_lin: res_stamp
n1: 1 n2: 2
resistance: 10

Ind
type: Linear
stamp_lin: ind_stamp
update: ind_update
n1: 2 n2: 3
L: 1e-3 dt: 5e-6
i_prev: 0
v_prev: 0

Cap
type: Linear
stamp_lin: cap_stamp
update: ind_update
n1: 3 n2: 0
C: 1e-6 dt: 5e-6
i_prev: 0
v_prev: 0

Input Parsing

We support the definition of:

- Voltage and current sources
- Voltage controlled voltage and current sources
- Resistors
- Capacitors
- Inductors
- Diodes
- NMOS and PMOS

Miscellaneous

- .TRAN (time information for transient analysis)
- Comments

Diode

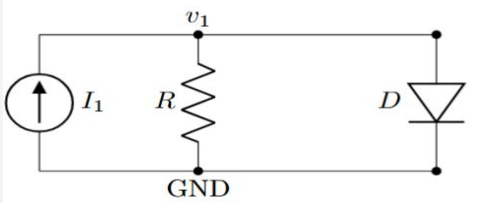
Newton's method

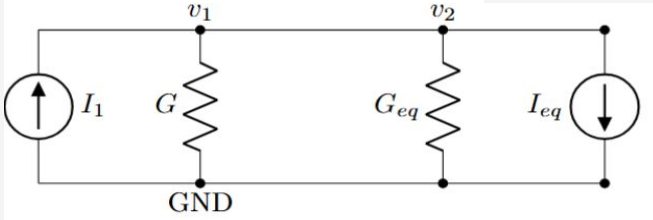
$$i_d = I_s(e^{v_d/V_t} - 1)$$

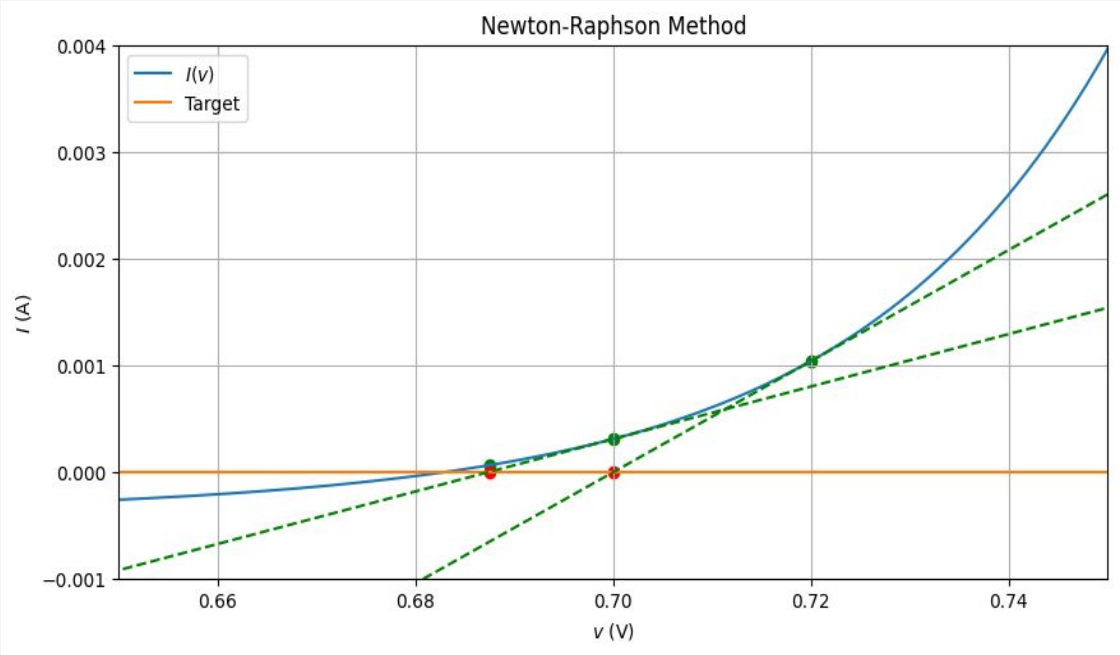
$$G_{eq} \equiv \frac{di_d}{dv_d} = \frac{I_s}{V_t} e^{v_{d0}/V_t}$$

$$\begin{aligned} i_d &\approx i_{d0} + G_{eq}(v_d - v_{d0}) \\ &= (i_{d0} - G_{eq} v_{d0}) + G_{eq} v_d \end{aligned}$$

1. Stamp
2. Solve
3. Update







Capacitor

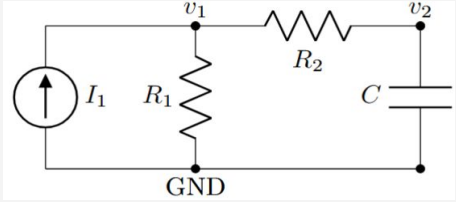
Make a time step approximation

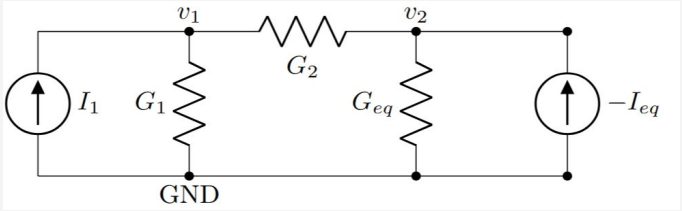
$$i(t) = C\frac{dv_C}{dt} = C\frac{d(v_2 - 0)}{dt}$$

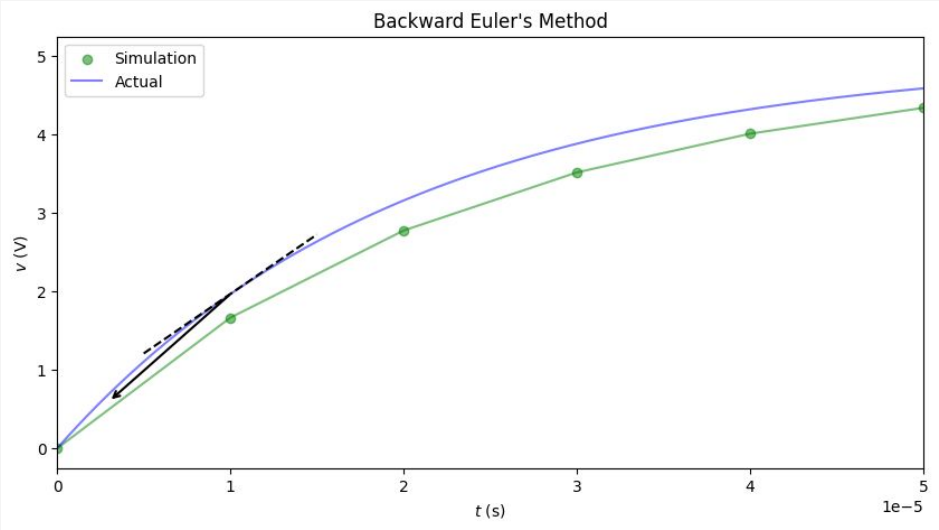
$$i(t_0 + \Delta t) \approx C\frac{v_2(t_0 + \Delta t) - v_2(t_0)}{\Delta t}$$

$$v_2(t_0 + \Delta t) = v_2(t_0) + \frac{\Delta t}{C} i(t_0 + \Delta t)$$

$$v_2(t) = v_2(t_0) + \frac{\Delta t}{C} i(t)$$

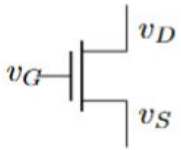


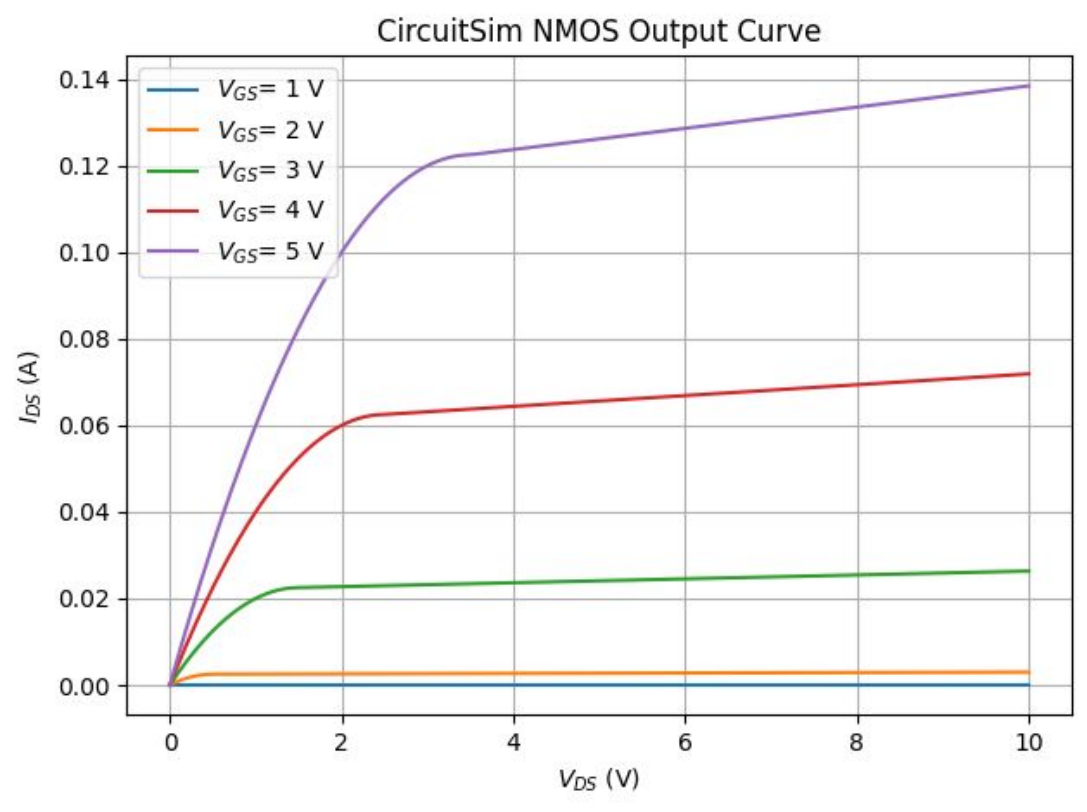




MOSFETs

Piecewise nonlinear equation
Multivariable linear models





Matrix Solver

For each timestep:

- Stamp static components
- While (not converged)
  - Stamp linear components
  - Copy *G* and **I** to FPGA
  - `ioctl()` Solves matrix on hardware
  - Read **v** from FPGA

Theory

Gaussian Elimination:

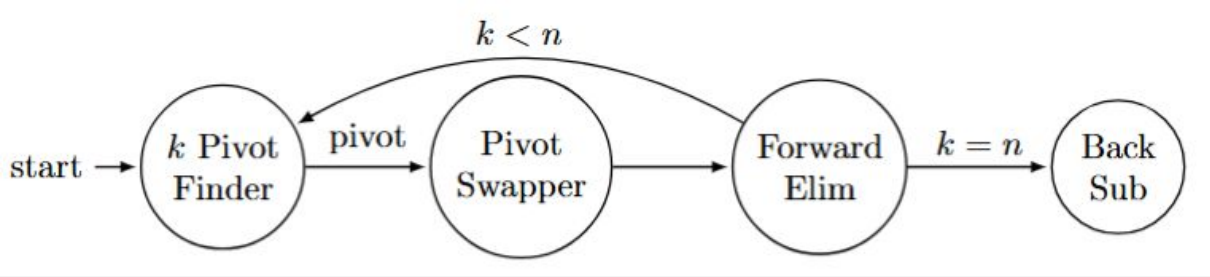
$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix}$$

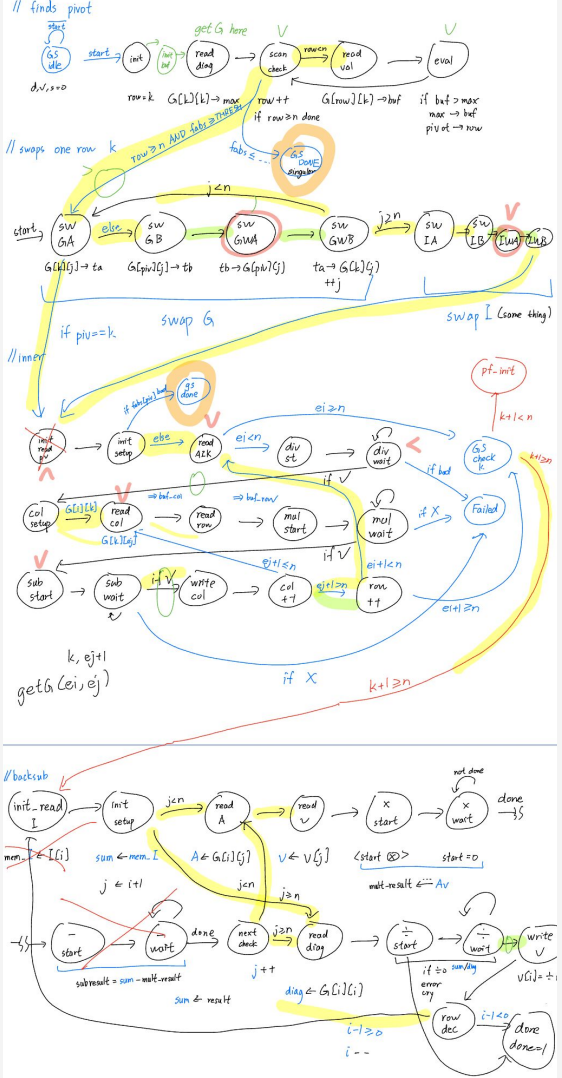
$$\begin{bmatrix} -2 & 1 & -1 \\ 1 & -2 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} -2 & 1 & -1 \\ 0 & -1.5 & 0.5 \\ 0 & -0.5 & -0.5 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} -2 & 1 & -1 \\ 0 & -1.5 & 0.5 \\ 0 & 0 & -2/3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 14/3 \end{bmatrix}$$

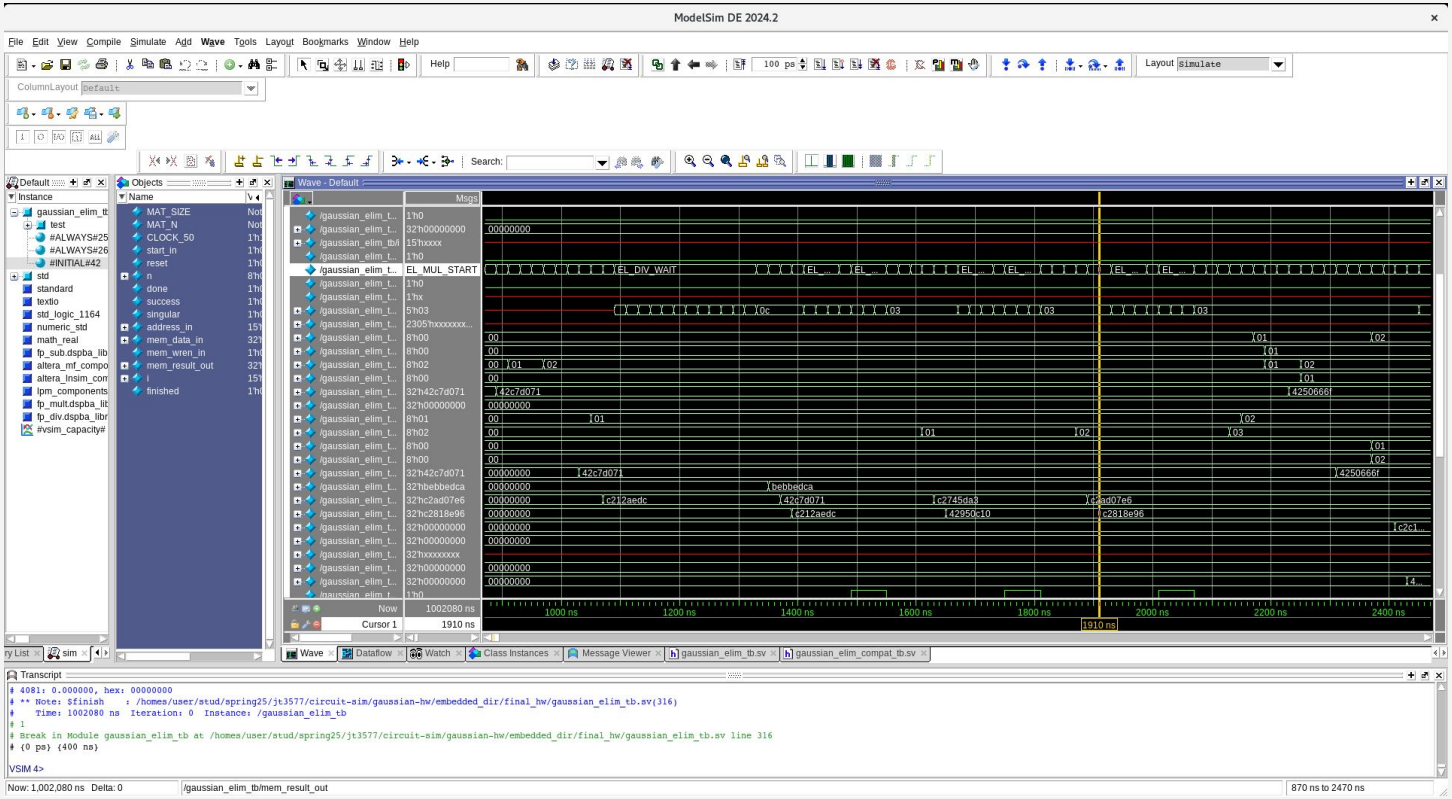
State Machine



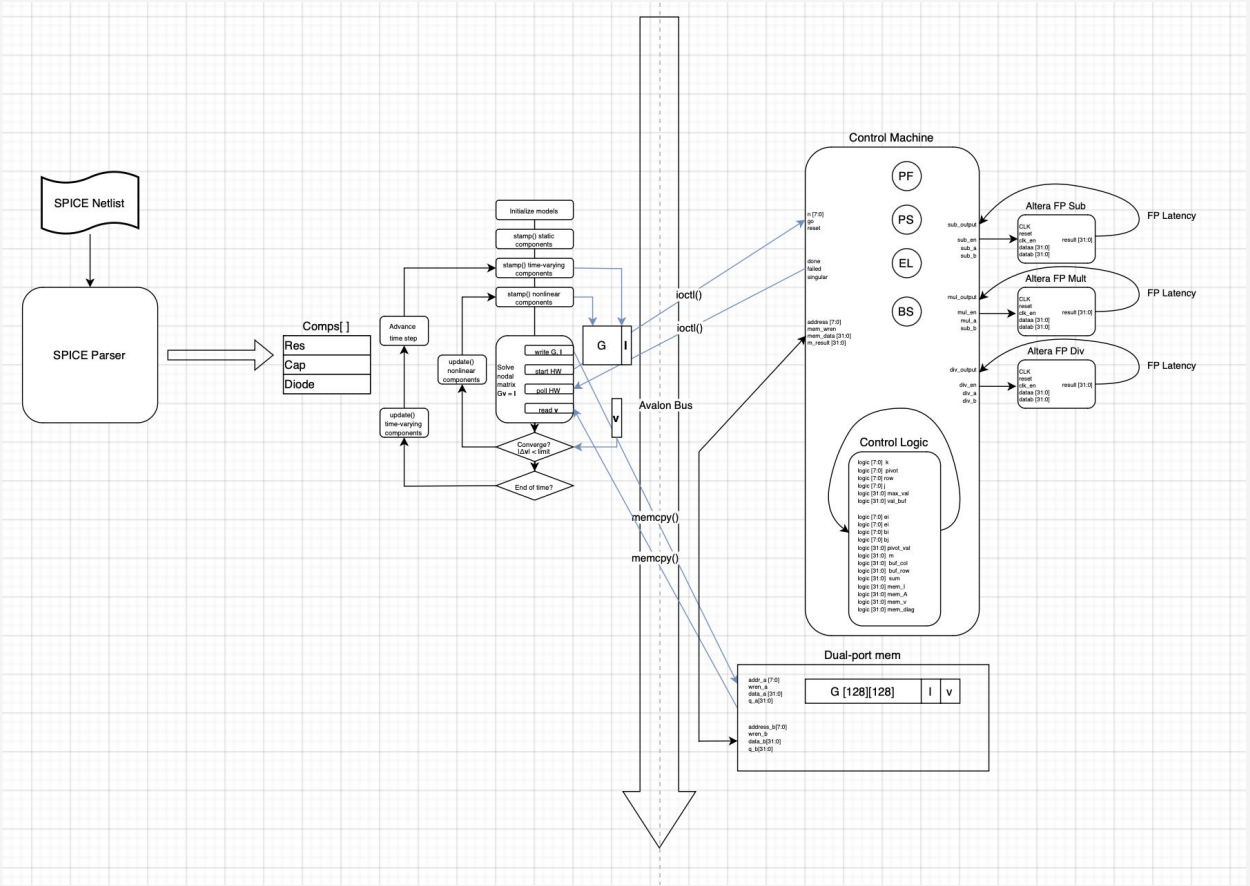


gaussian_elim

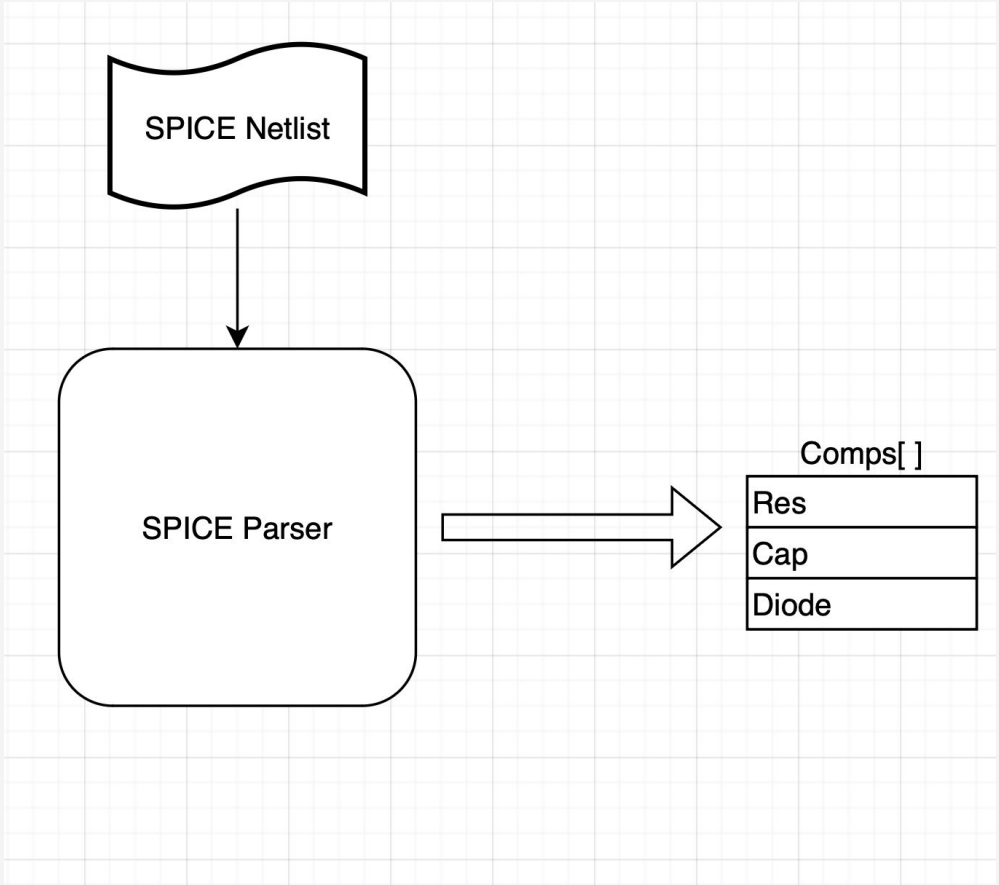
Memory access has two cycle delay, pipelining
Custom FP operations to speed system up

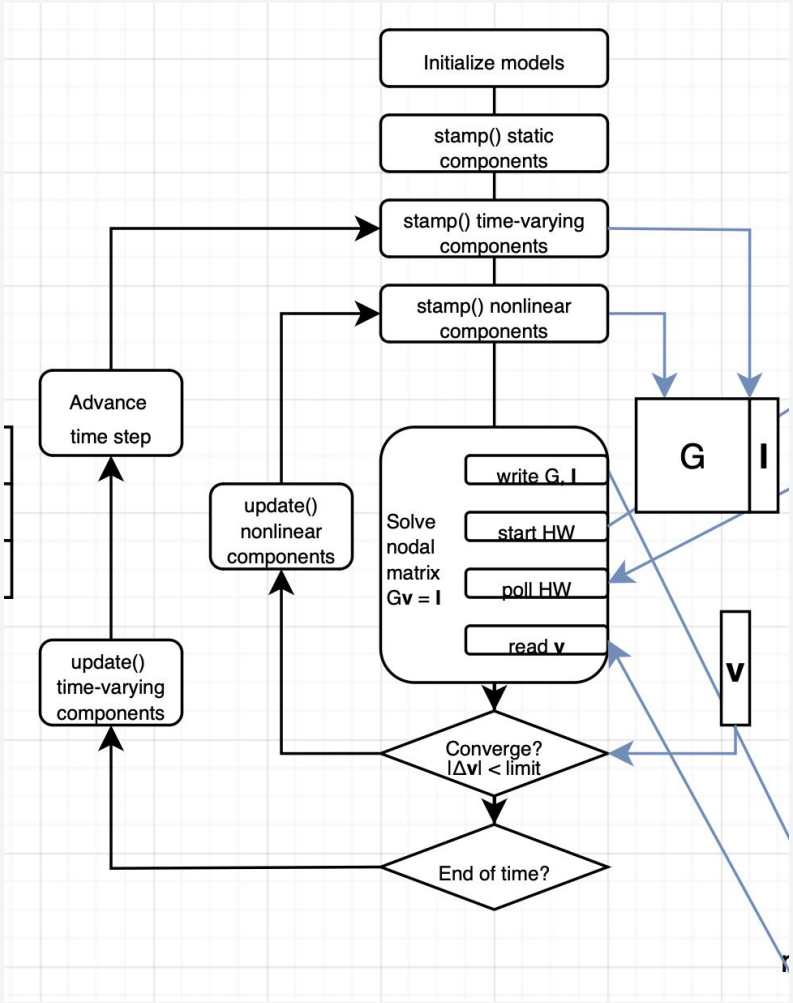


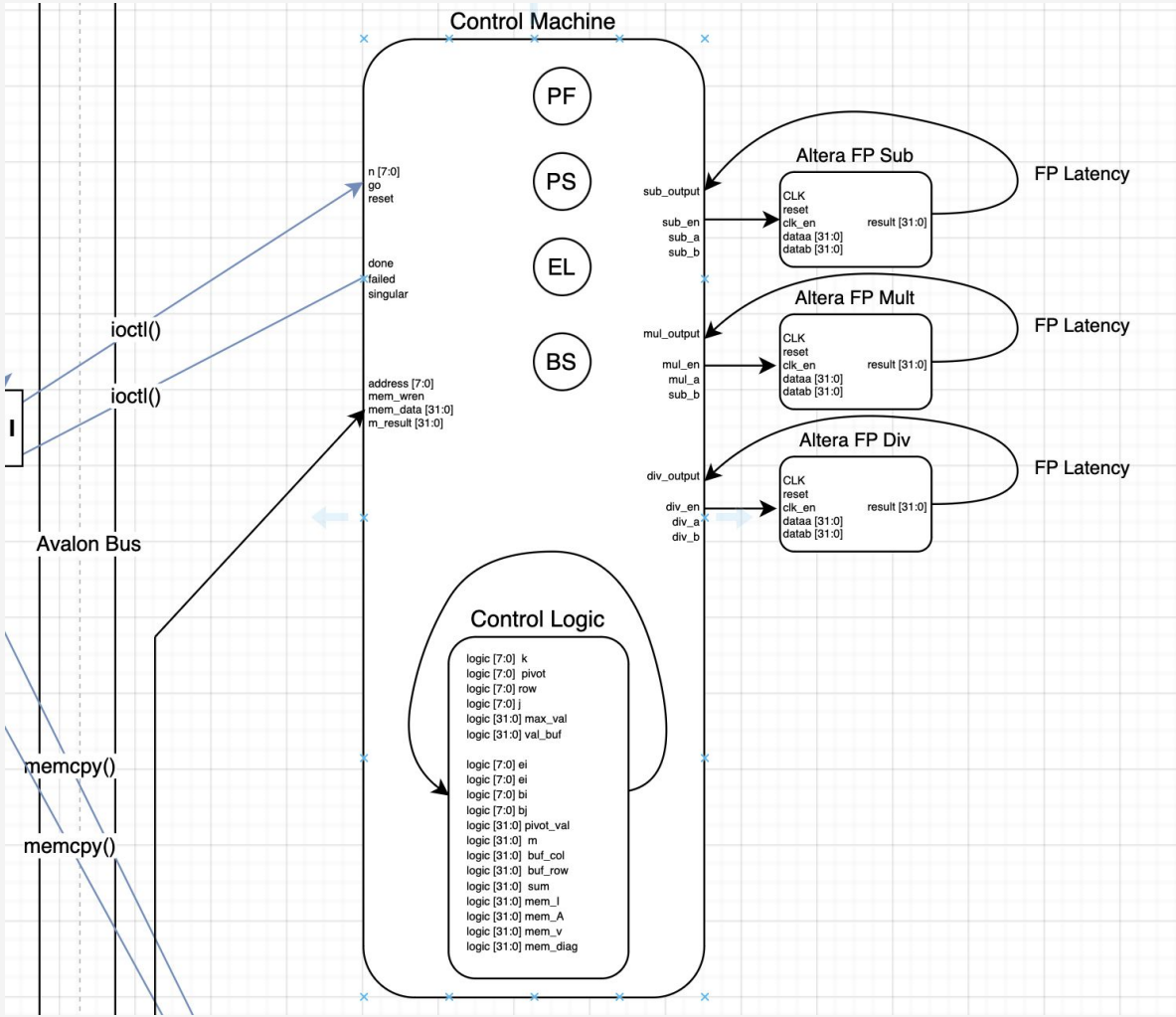
State Block Diagram

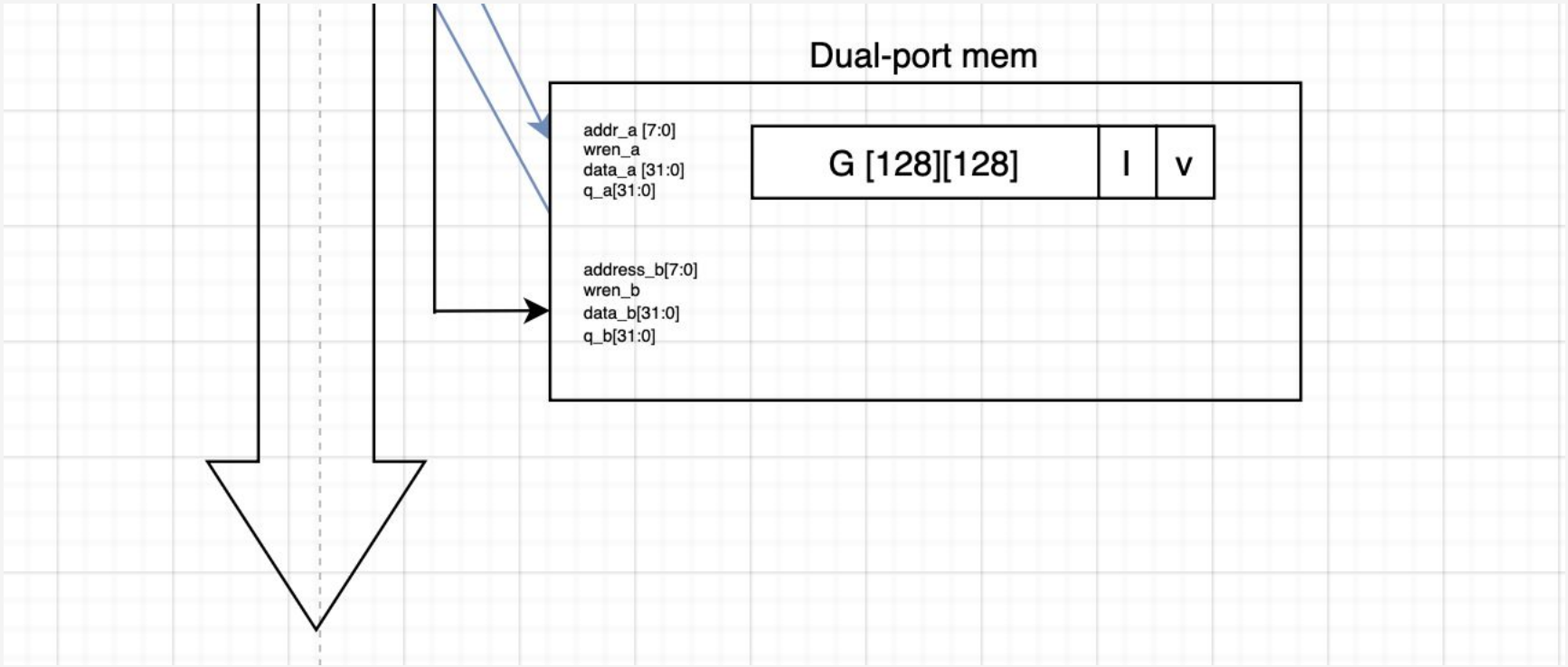


State Block Diagram




Initialize models
stamp() static components
stamp() time-varying components
stamp() nonlinear components
Advance time step
update() nonlinear components
update() time-varying components
Solve nodal matrix Gv = I
write G, I
start HW
poll HW
read v
G
I
v
Converge? |Δv| < limit
End of time?

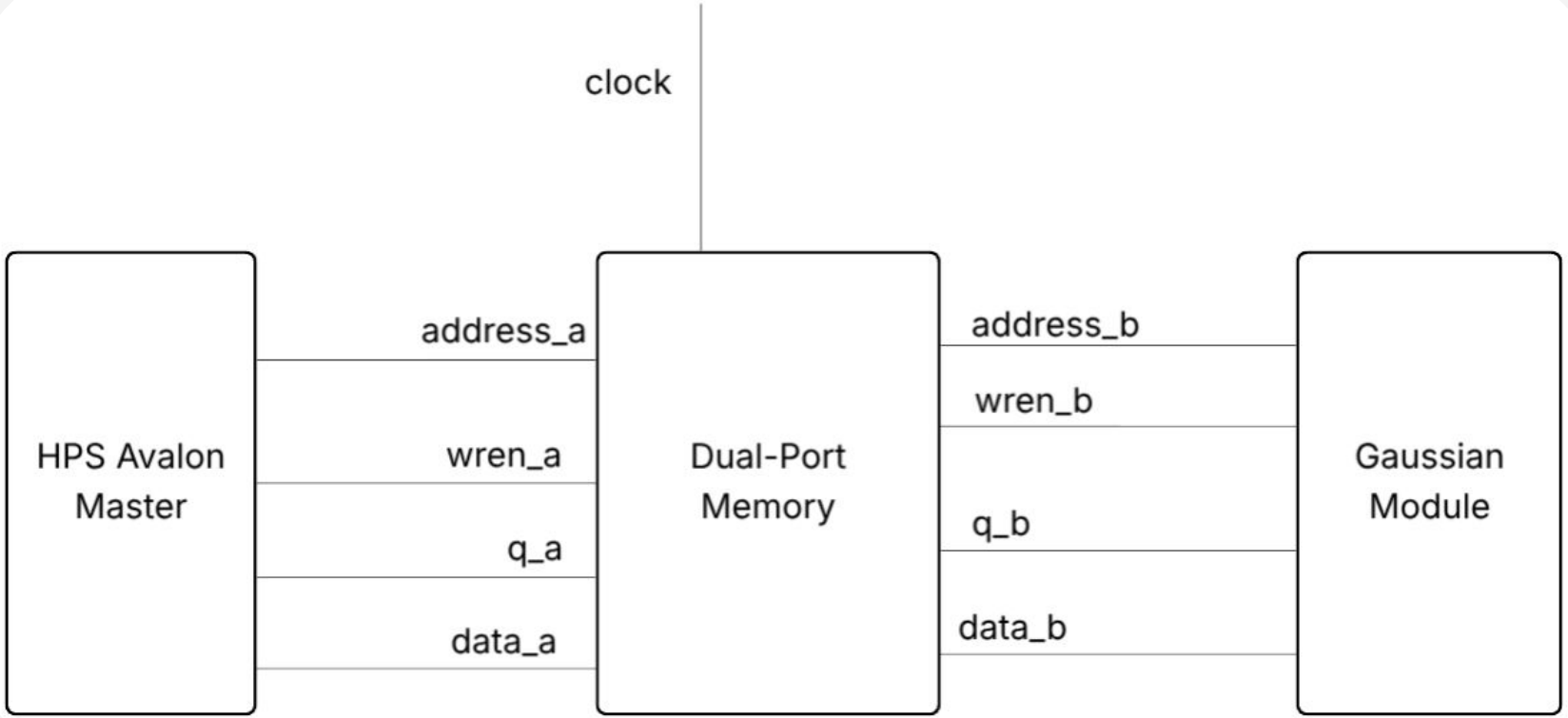

Control Machine
PF
PS
EL
BS
n [7:0]
go
reset
done
failed
singular
address [7:0]
mem_wren
mem_data [31:0]
m_result [31:0]
ioctl()
ioctl()
Avalon Bus
memcpy()
memcpy()
Altera FP Sub
sub_output
sub_en
sub_a
sub_b
CLK
reset
clk_en
dataa [31:0]
datab [31:0]
result [31:0]
FP Latency
Altera FP Mult
mul_output
mul_en
mul_a
sub_b
FP Latency
Altera FP Div
div_output
div_en
div_a
div_b
FP Latency
Control Logic
logic [7:0] k
logic [7:0] pivot
logic [7:0] row
logic [7:0] j
logic [31:0] max_val
logic [31:0] val_buf
logic [7:0] ei
logic [7:0] ei
logic [7:0] bi
logic [7:0] bj
logic [31:0] pivot_val
logic [31:0] m
logic [31:0] buf_col
logic [31:0] buf_row
logic [31:0] sum
logic [31:0] mem_l
logic [31:0] mem_A
logic [31:0] mem_v
logic [31:0] mem_diag


Dual-port mem
addr_a [7:0]
wren_a
data_a [31:0]
q_a[31:0]
G [128][128]
I
v
address_b[7:0]
wren_b
data_b[31:0]
q_b[31:0]

FP IP

Intel FPGA, set FP to work for 50MHz clock and had corresponding latencies

M10K Memory


clock
HPS Avalon Master
address_a
wren_a
q_a
data_a
Dual-Port Memory
address_b
wren_b
q_b
data_b
Gaussian Module

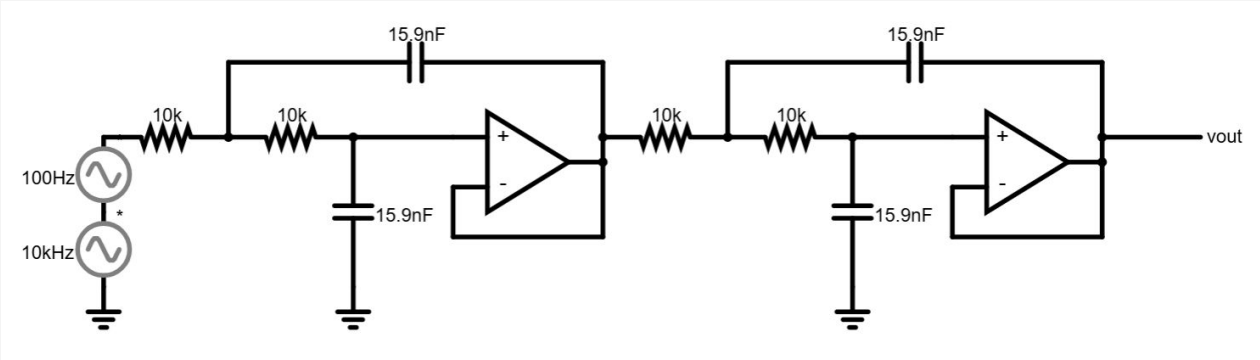
Control Signal `ioctl()`

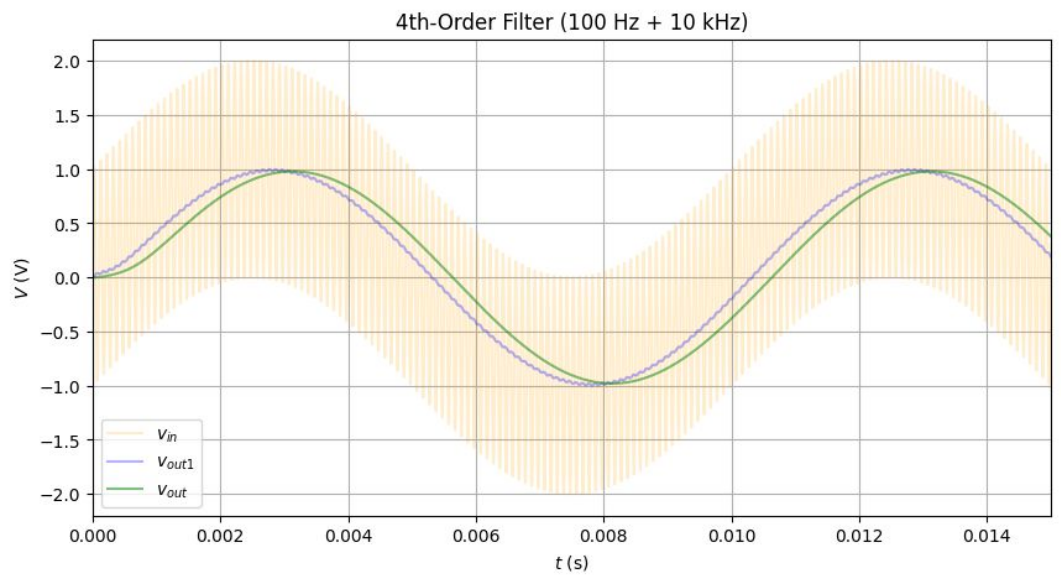
Matrix dim (8 bits)			
G	R	Unused (6 bits)	
D	E	S	Unused (5 bits)

Platform Designer

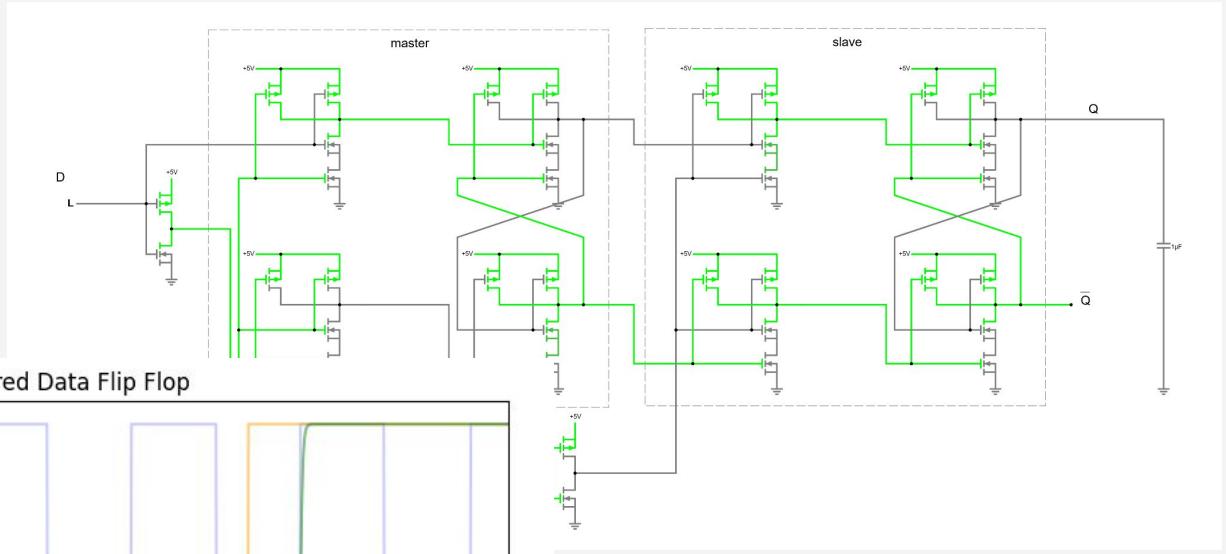
Use	Connections	Name	Description	Export	Clock	Base	End
☑		⊟ **clk_0**	Clock Source				
		clk_in	Clock Input	**clk**	***exported***		
		clk_in_reset	Reset Input	**reset**			
		clk	Clock Output	*Double-click to*	clk_0		
		clk_reset	Reset Output	*Double-click to*			
☑		⊟ **hps_0**	Arria V/Cyclone V Hard Proce...				
		h2f_user1_clock	Clock Output	*Double-click to*	hps_0_h2...		
		memory	Conduit	**hps_ddr3**			
		hps_io	Conduit	**hps**			
		h2f_reset	Reset Output	*Double-click to*			
		h2f_axi_clock	Clock Input	*Double-click to*	**clk_0**		
		h2f_axi_master	AXI Master	*Double-click to*	[h2f_axi_...		
		f2h_axi_clock	Clock Input	*Double-click to*	**clk_0**		
		f2h_axi_slave	AXI Slave	*Double-click to*	[f2h_axi_...		
		h2f_lw_axi_clock	Clock Input	*Double-click to*	**clk_0**		
		h2f_lw_axi_master	AXI Master	*Double-click to*	[h2f_lw_a...		
☑		⊟ **gaussian_0**	gaussian				
		clock	Clock Input	*Double-click to*	**clk_0**		
		reset	Reset Input	*Double-click to*	[clock]		
		csr	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000_0000**	0x0000_0003
		mem	Avalon Memory Mapped Slave	*Double-click to*	[clock]	**0x0000 0000**	0x0001 ffff

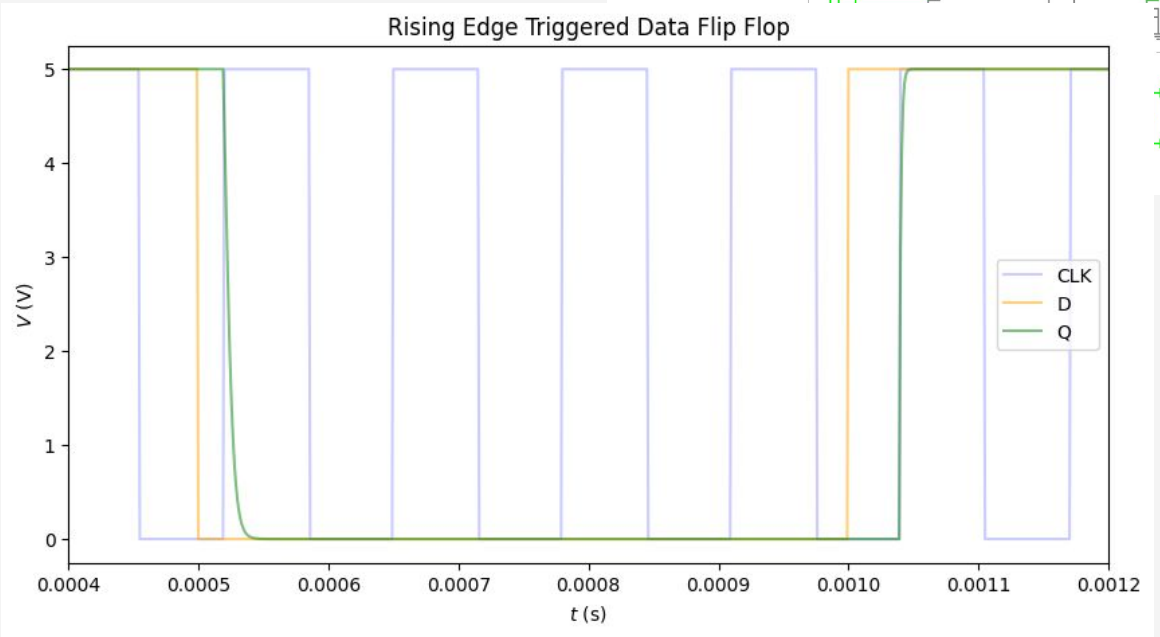
4th-Order Filter



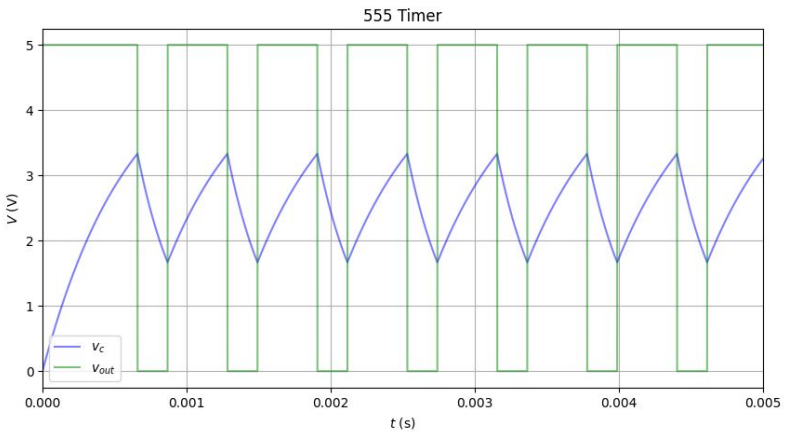


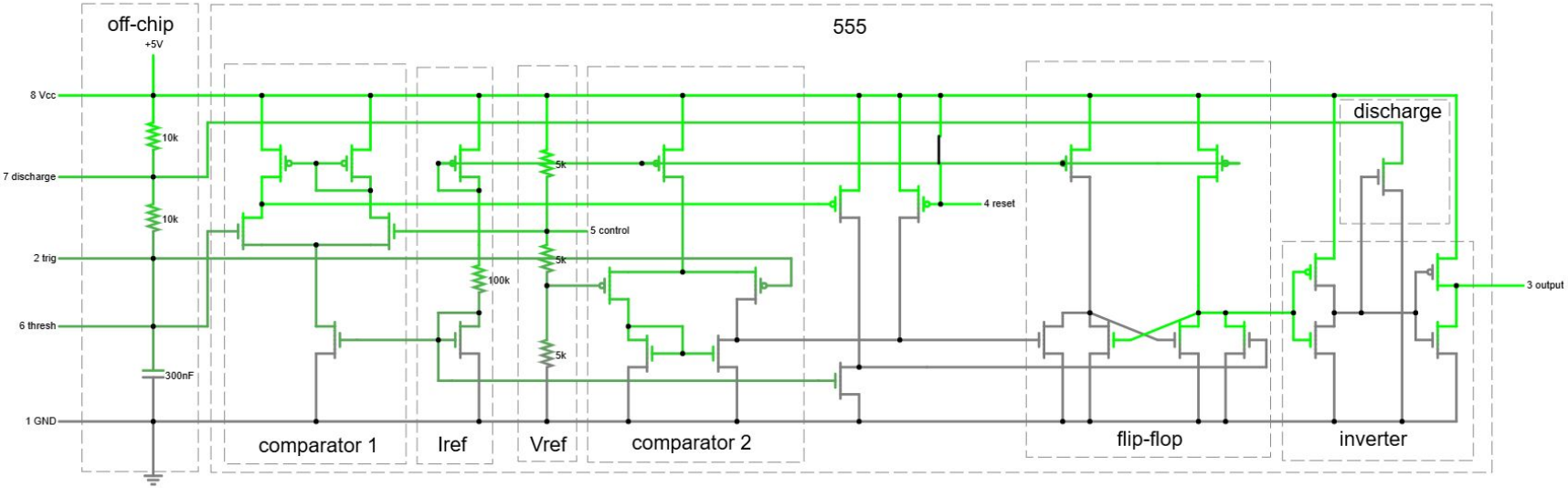
CMOS


master
slave
D
L
Q
$\overline{Q}$
+5V
1µF

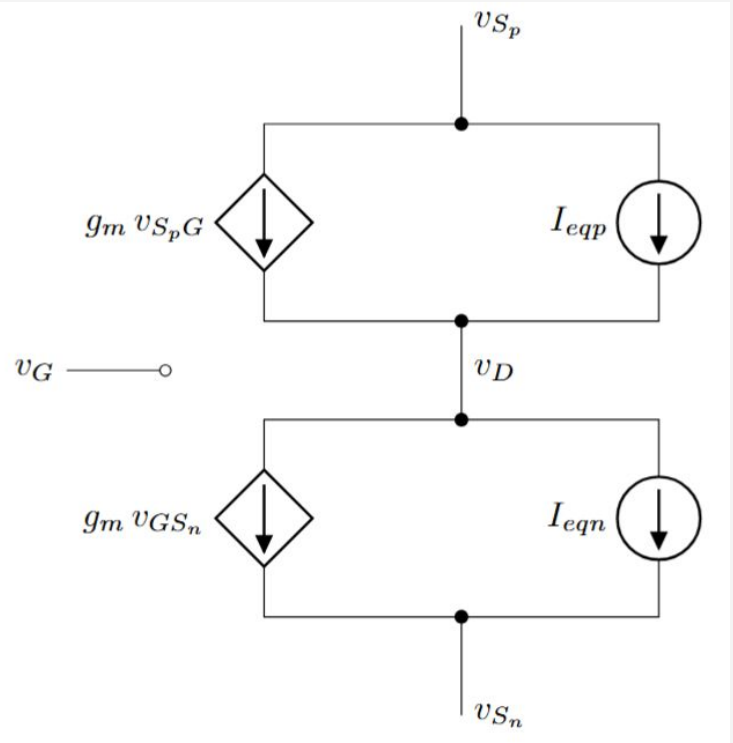

Rising Edge Triggered Data Flip Flop
V (V)
t (s)
CLK
D
Q
0
1
2
3
4
5
0.0004
0.0005
0.0006
0.0007
0.0008
0.0009
0.0010
0.0011
0.0012

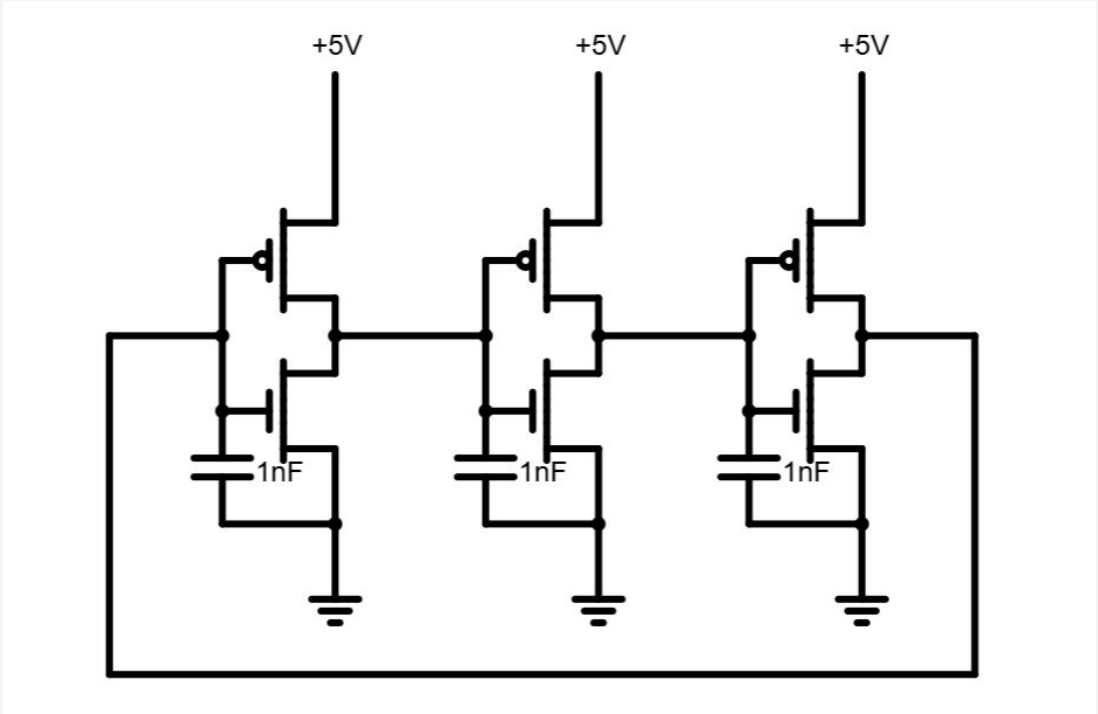
555 Hybrid Chip





Failure Modes





Acknowledgements

Stephen Edwards

Peiran Wang

https://www.h-schmidt.net/FloatConverter/IEEE754.html