# Closed Loop Digital LDO Linear Controller

*Zinia Tuli*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-137
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-137.html

October 2, 2019

# Closed Loop Digital LDO Linear Controller

Zinia Tuli

9/27/2019

# Agenda

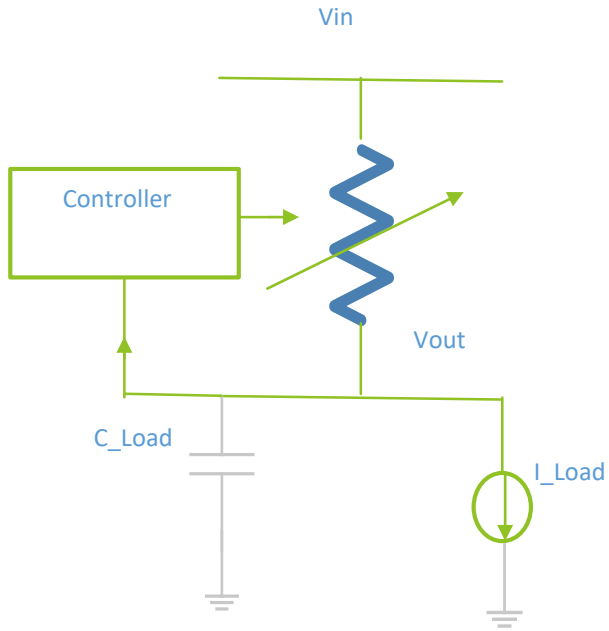
- Problem statement
- Basic LDO
- Architecture
- Test bench
- Results
- Details of Closed Loop LDO
- Discussion
- Challenges

# Problem Statement

Want to build a closed loop digital LDO linear controller that provides constant Vout regardless of variable Vin with

1. At least 75% current efficiency when I_Load=0.35mA
2. 100mV regulation range
3. High clock frequency
4. Stable/Constant Vout

# Basic LDO

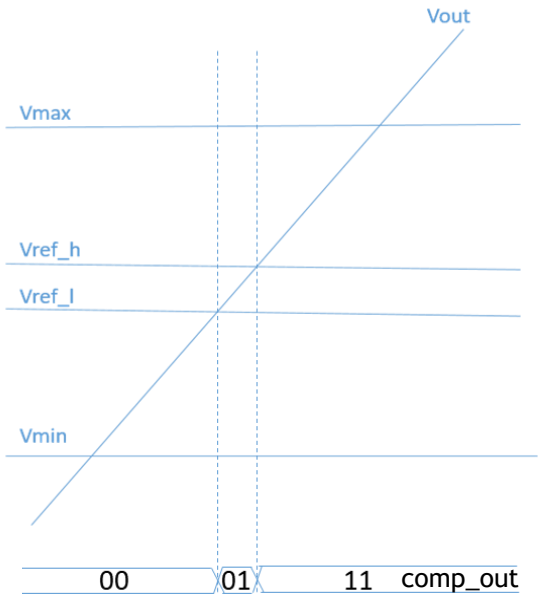



- Low dropout regulators (LDOs) are effective ways to regulate an output voltage that is powered from a higher input voltage
- A basic LDO involves a variable resistor (can be implemented by using PMOS transistors), I_Load, C_Load, and a controller.
- The job of the controller is to keep the output voltage constant even with variable input voltage. If the load current (I_Load) increases, the variable resistor should decrease to increase the current flow from the input through the resistor to the output and vice versa in order to make Vout remain constant.
- The controller detects the deviation of the Vout from its desired value and responds accordingly to either increase or decrease the variable resistor's value to keep the Vout constant.

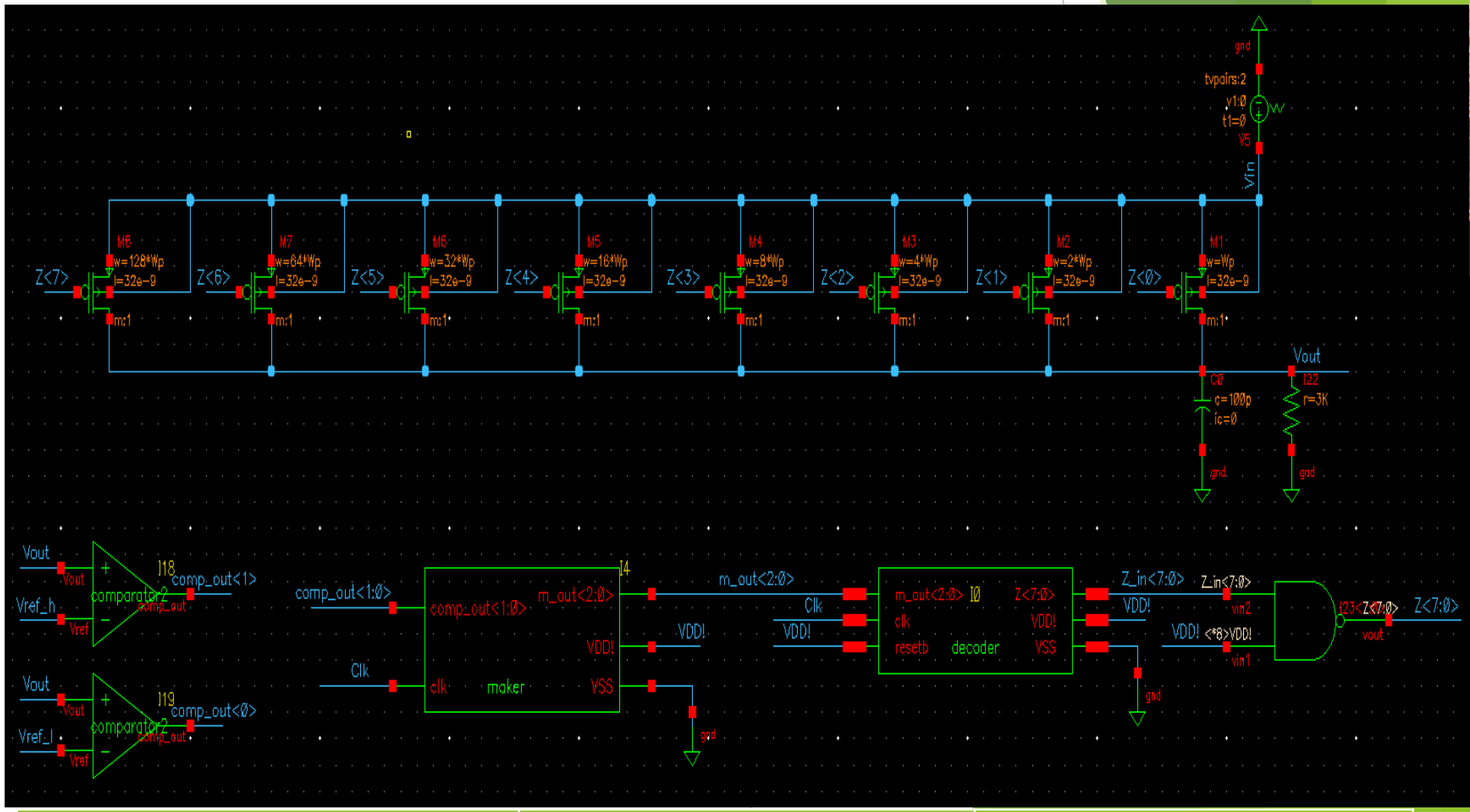
# Architecture

Closed loop design of digital LDO with digital controller (Decision maker and Decision executer), 2 Comparators and Pmos devices (binary coding power gates which has 8 different size), CL, IL:

## Input of Comparators:




| comp_out | m_out | Z |
|---|---|---|
| 00=0 | Up=100=4 | Count up |
| 11=3 | Down=010=2 | Count down |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |




| comp_out (output of comparators) | m_out (output of decision maker) | Z (output of decision executer) |
|---|---|---|

# Continued….

- The job of two comparators is to detect the position of Vout and feed the information into the decision maker block.
- The decision maker block takes that information and decides if Vout should go up, down, or stay hold within Vref_h and Vref_l.
- The decision maker feeds this information into the 8 bit binary counter or decision executer which physically changes the value of power gate code <7:0> (input of the Pmos devices) and thus Vout.
- An example: when Comp_h=0 (Vref_h> Vout, or Vout is lower) and Comp_l=0 (Vref_l> Vout, or Vout is lower)→Decision maker output=Up→Decision executer output = Power gate code<7:0> +1. Thus, Power gate code counts up from its previous value by turning on more Pmos devices.

Combined truth table of digital controller of LDO:

| comp_out (output of comparators) | m_out (output of decision maker) | Z (output of decision executer) |
|---|---|---|
| 00=0 | Up=100=4 | Count up: Power gate code<7:0> +1 |
| 11=3 | Down=010=2 | Count down: Power gate code<7:0> -1 |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |

# Continued….

- Pmos device unit size: W=210nm, L=32nm
- Vin=1.05V
- C_Load=200pF

- Proposed Metrics:
    1. Current efficiency= 75-99% for 0.35mA current load (RL=3KΩ)
    2. Regulation range= 100mV
    3. Clk frequency=1MHz – 1GHz
    4. Vout being stable within regulation range

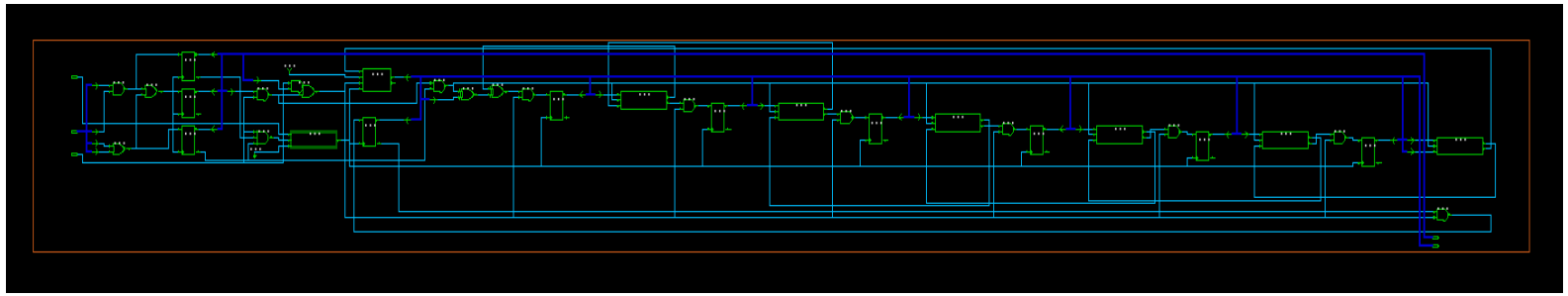
# Testbench and RTL code

```
`timescale 1 ns /  1 ps
`define expect(nodeName, nodeVal, expVal, cycle) if (nodeVal !== expVal) begin \
  $display("\t ASSERTION ON %s FAILED @ CYCLE = %d, 0x%h != EXPECTED 0x%h", \
  nodeName,cycle,nodeVal,expVal); $stop; end

module decoder_tb();
  reg [1:0] comp_out = 0;
  wire [2:0] m_out;
  wire [7:0] Z;
  reg clk = 0;
  wire resetb;
  integer cycle = 0;
  integer i;
wire up, down, hold;
assign up = m_out ==3'b100;
assign down = m_out ==3'b010;
assign hold = m_out ==3'b001;

  always #(`CLOCK_PERIOD * 0.5)  clk = ~clk;
  always @(posedge clk) cycle = cycle + 1;
  decoder decoder_inst (.comp_out(comp_out), .m_out(m_out), .Z(Z), .clk(clk), .resetb(resetb));

reg [6:0] counter ;
initial begin
counter = 7'd0;
end

always @ (posedge clk)
begin
counter <= counter + 1 ;
end

assign resetb  =  (counter == 7'd0) ? 0 : 1;

  initial begin
    $vcdpluson;
    // Extra cycles to flush out any initial X's
    repeat(4) @(posedge counter[2]);
    for (i=1; i<=50; i=i+1) begin
      // Input + output events on clock rising edge
      @(posedge counter[2]);
      #0.001;
      comp_out = i;
    end
    $vcdplusoff;
    $display("\t **Ran through all test vectors**"); $finish;
  end

  initial begin
    $monitor($time,": comp_out=%d, m_out=%b, Z=%b, cycle=%d", comp_out, m_out, Z, cycle);
  end
endmodule
```

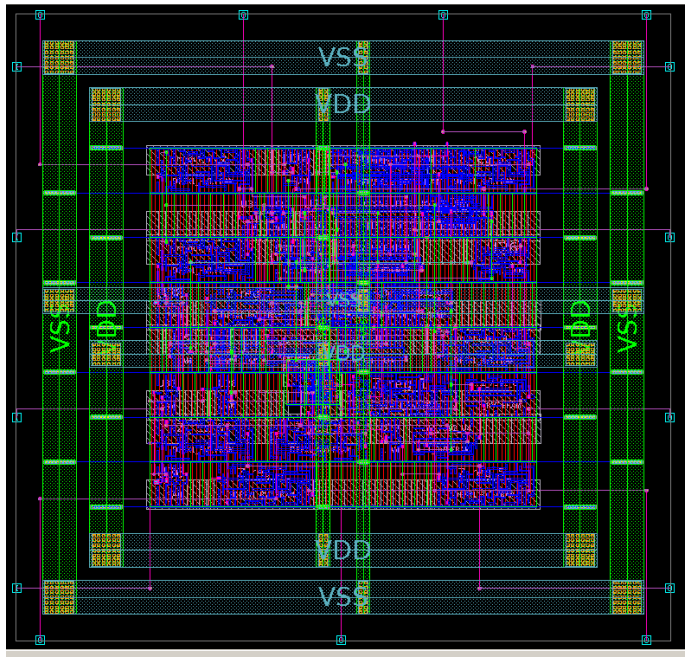
```
module decoder(
input [1:0] comp_out, //comp_out
input clk,
input resetb,
output reg [2:0] m_out,  //decision maker output (up down hold)
output reg [7:0] Z);     //decision executer output or power gate code

//decision maker
always @ (posedge clk)
begin
        if (comp_out ==2'b00) m_out=3'b100;             //output=up
        else if (comp_out == 2'b11) m_out=3'b010;       //output=down
        else m_out=3'b001;                              //when comp_out==2'b01 or 2'b10 output=hold
end

//decision executer
always @ (posedge clk)
begin
        if (!resetb) Z<=0;
        else if (m_out[2]) Z<=Z+1; //count up
        else if (m_out[1]) Z<=Z-1; // count down
        else Z<=Z; //steady state
end

endmodule
```

# Results

Gate Level Schematic of LDO controller (using IC compiler):

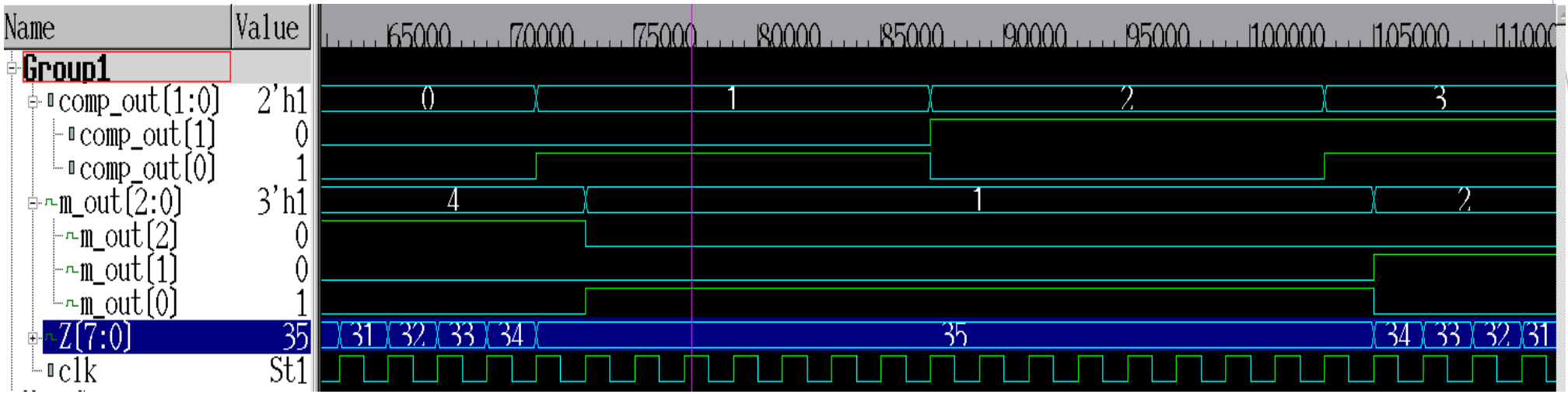


Layout of LDO controller(using IC compiler):



# Continued….

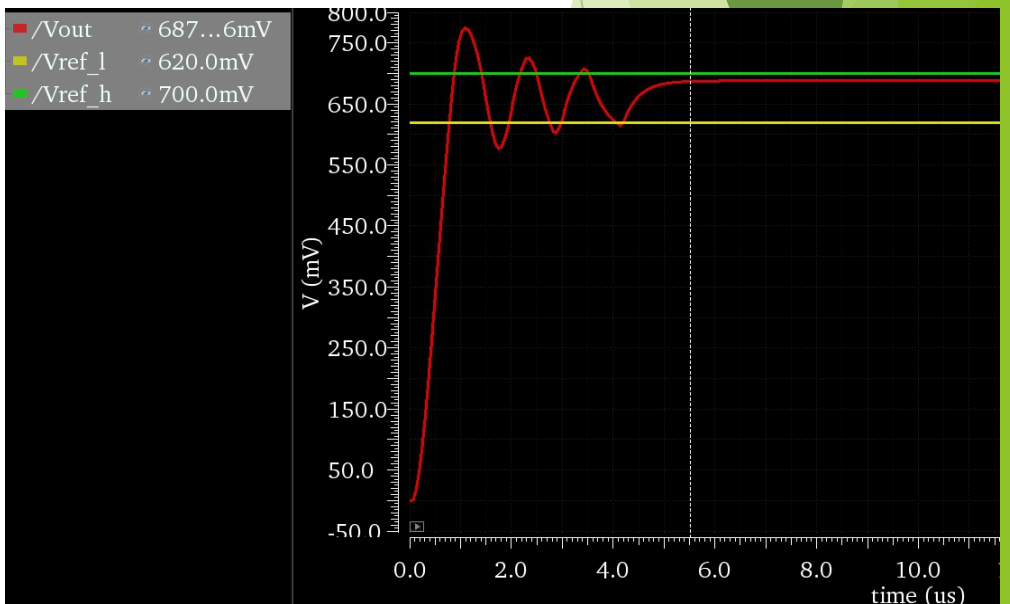
## Output Waveform (using DVE) of Digital LDO Controller:




In figure above, when Vout is between Vref_l & Vref_h, comp_out=01=**1**, m_out is Hold or 001=**1**, and Z is at steady state=**35** which means the controller works and when used in the closed loop LDO, the Vout looks like the graph on the right where Vout is constant/stable between Vref_h and Vref_l.

| comp_out | m_out | Z |
|---|---|---|
| 00=0 | Up=100=4 | Count up |
| 11=3 | Down=010=2 | Count down |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |

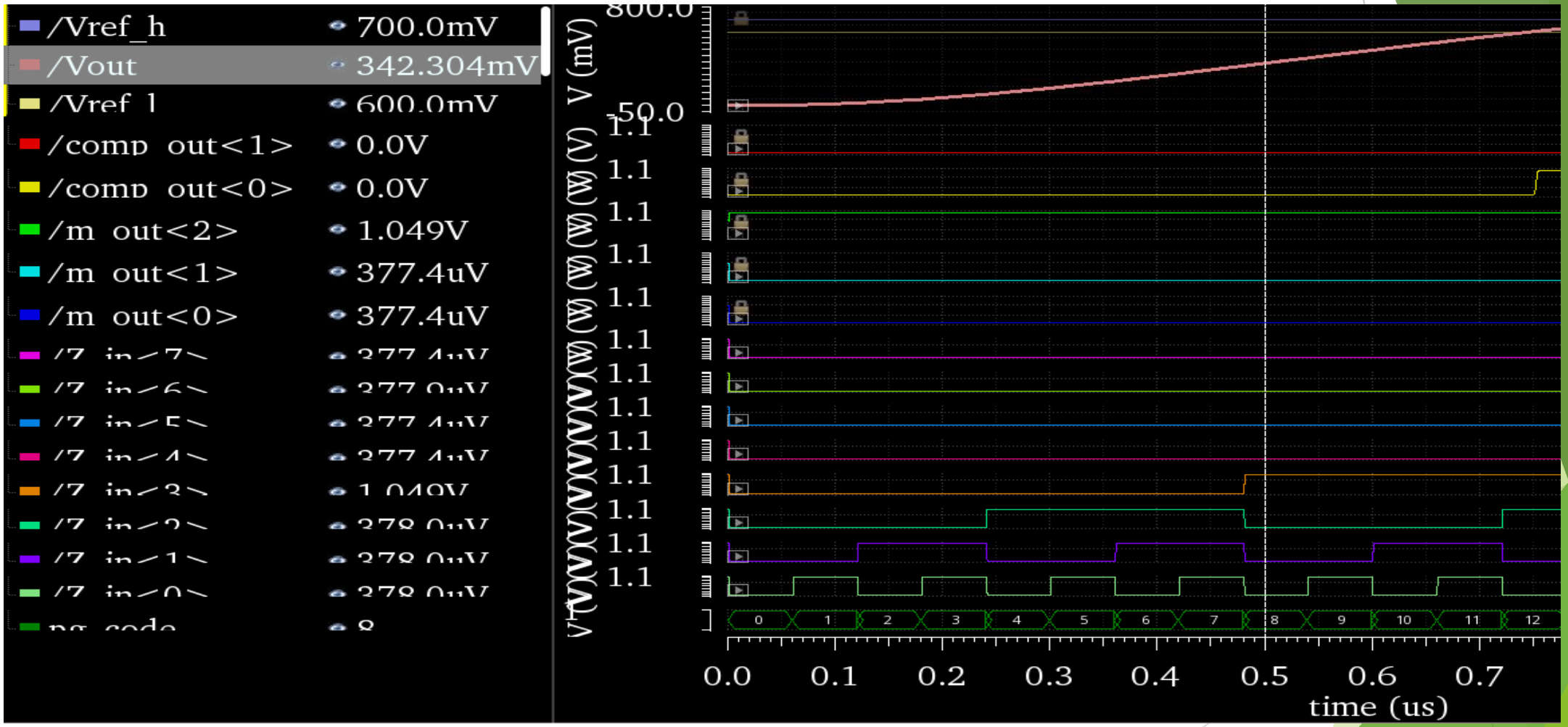
Vout waveform:The transient of the number of turned-on switches



| | | |
|---|---|---|
| 00=0 | Up=100=4 | Count up |
| 11=3 | Down=010=2 | Count down |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |

# Details of Closed Loop LDO

1. Comparator, decision maker, decision executer output waveform when Vout is increasing at 0.5us:

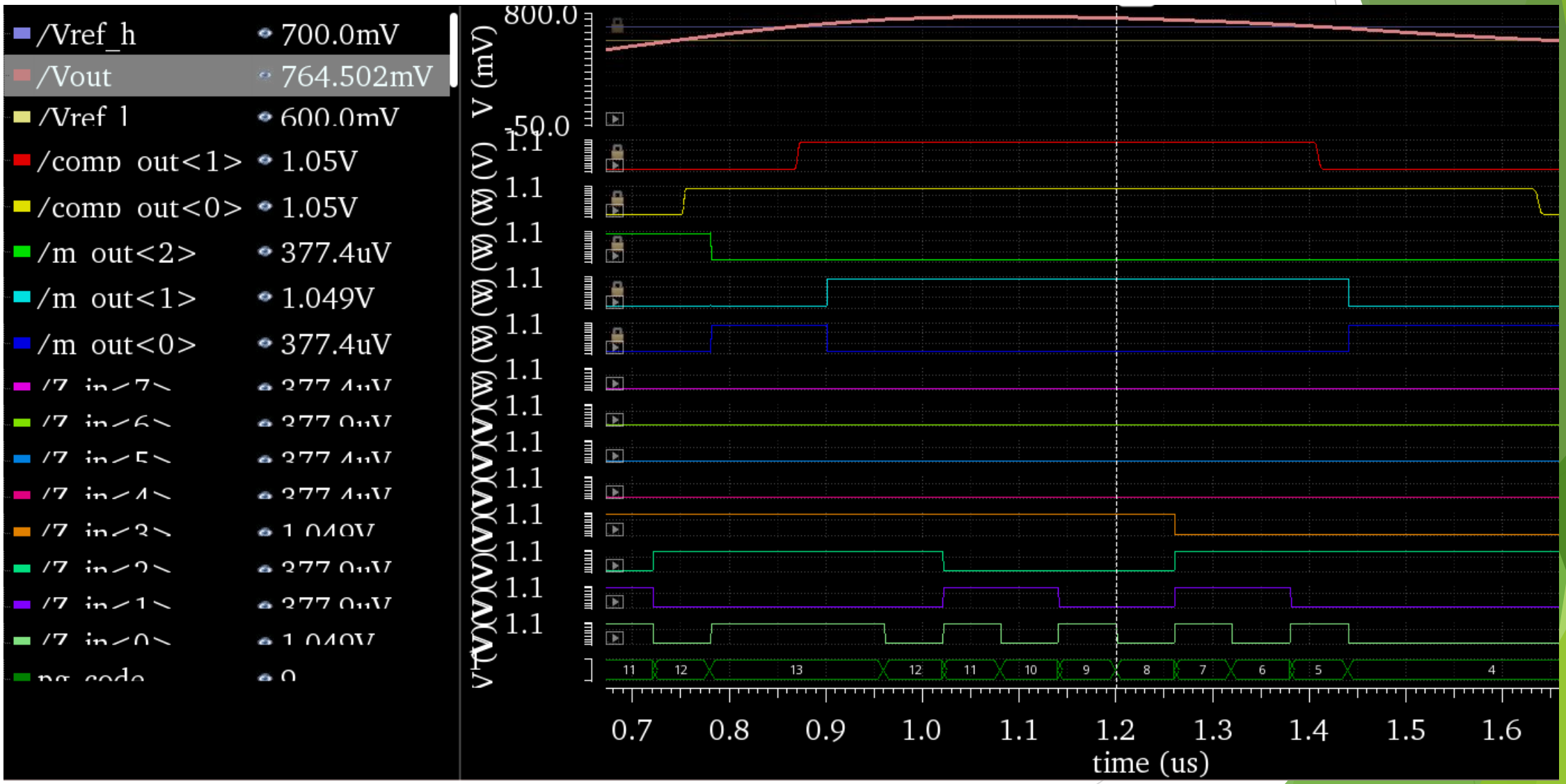



Comp_out=00, m_out=100 (Up), pg code is counting up from 7 to 8 to higher # and therefore Vout is increasing

# Continued….

| | | |
|---|---|---|
| 00=0 | Up=100=4 | Count up |
| 11=3 | Down=010=2 | Count down |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |

2. Comparator, decision maker, decision executer output waveform when Vout is decreasing at 1.2us:

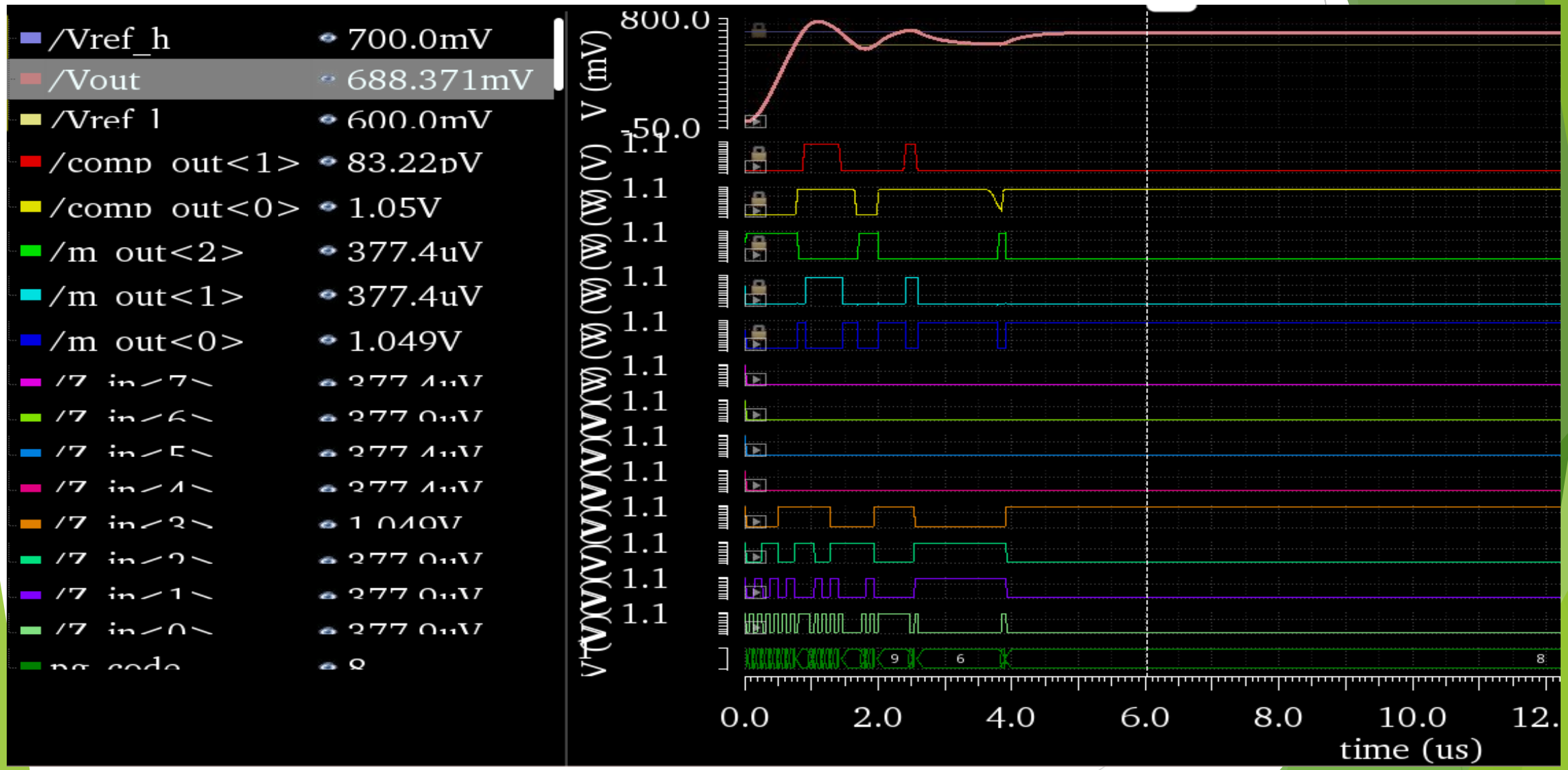



Comp_out=11, m_out=010 (Down), pg code is counting down from 9 to 8 to lower # and therefore Vout is decreasing

| | | |
|---|---|---|
| 11=3 | Down=010=2 | Count down |
| 01=1 | Hold=001=1 | Steady state |
| 10=2 | Hold=001=1 | Steady state |

# Continued….

3. Comparator, decision maker, decision executer output waveform when Vout is steady at 6us:




Comp_out=01, m_out=001 (Hold), pg code is at steady state at 8 and therefore Vout is constant

# Discussion

1. Current Efficiency:
   - Using Invx1_rvt standard cells, clock frequency=111MHz, C_Load=0.01fF, supply voltage=1.05V power consumption of the digital controller was= 90uW after APR
   - The current leakage =power consumed by controller/Vin= 90uW/1.05V= 85.7uA. Therefore current efficiency=I_Load/(I_Load+I_Leak)=0.35mA/(0.35mA+.0857mA)=0.80 →80% which is within proposed metrics

2. Regulation Range:

   In this design I was able to achieve voltage range=80mV between Vmin=0V and Vmax=Vin. This turned out to be even better than expected value (100mV)

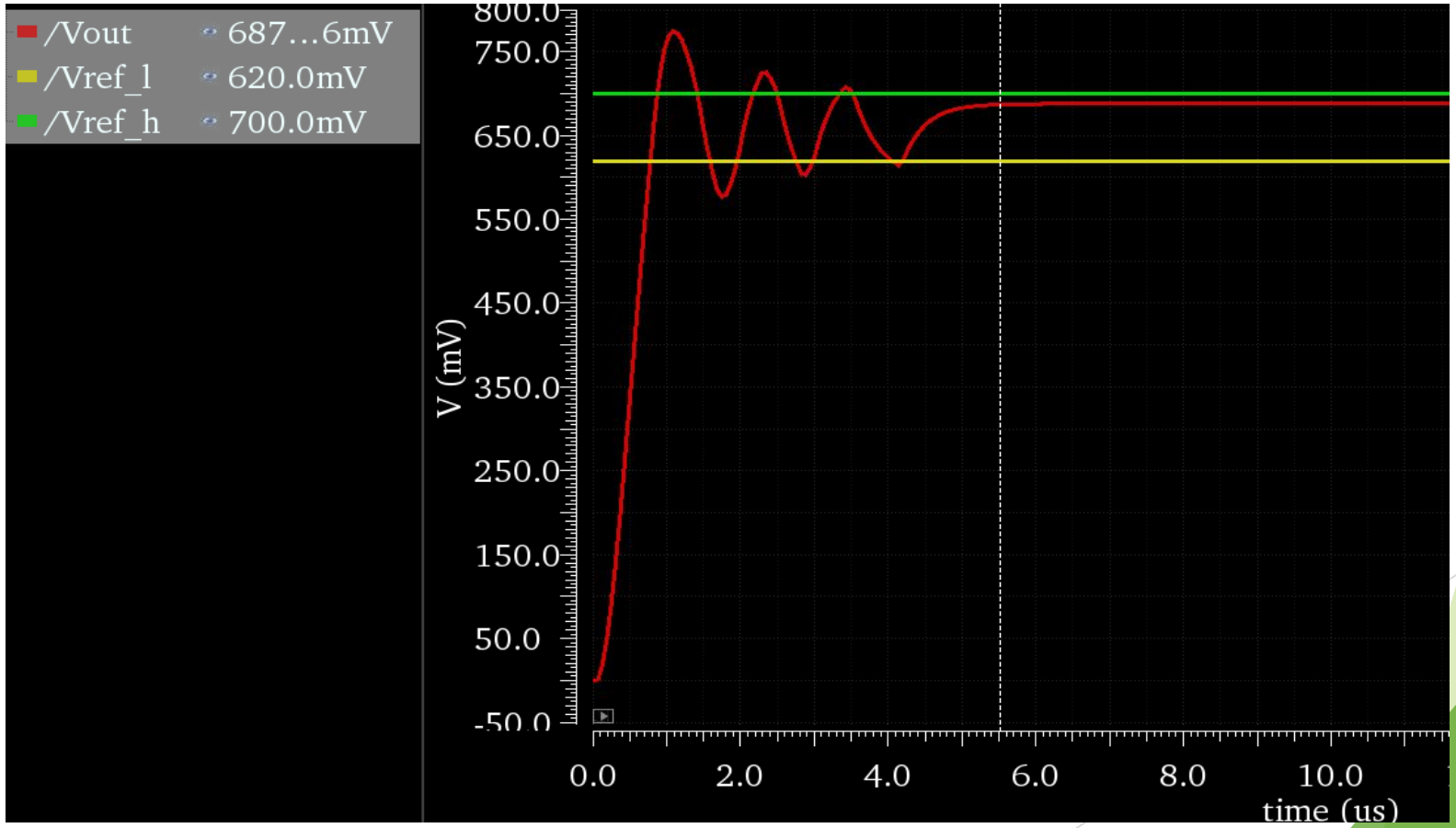
3. Clock Frequency:

   For the full closed loop LDO, clk period was=6ns→clk freq=0.17GHz which meets the proposed metrics

# Discussion Continued…

4. Vout Stability:

With the closed loop, the voltage profile looks like the figure below, where Vout is stable/constant between Vref_l=620mV and Vref_h=700mV.



# Challenges

1. Vout was increasing above Vref_l and Vref_h even though Z_in<7:0> was steady:

- The reason was the current of Vout was too low compared to the pmos gates. So the current of a single pmos at Vds=700mV was measured and multiplied by 128 (because the biggest pmos size is 128x) which was equal to ~30uA*128=3840uA=3.8mA. That means I_Load<3.84mA→RL>0.3KΩ. First RL=1KΩ was chosen but RL=3KΩ gave better Vout curve. The lower the I_Load the easier it is for the pmos gates to match the current with it.
- Still Vout was too high, so clk frequency was lowered by 3x and step size was decreased which made Vout to be in between Vref_l and Vref_h.

2. Vout was too rough and limit cycling (Vout is limited by the resolution of LSB) occurred:

- CL was increased by 2x (200pf) which made Vout smoother. Voltage range (between Vref_l and Vref_h) could have increased also to fix this issue.

# Thank you!

Questions???