Behavioral Modeling of I/O Drivers Using Neural Networks

Students: Huan Yu, Mourad Larbi (postdoc)

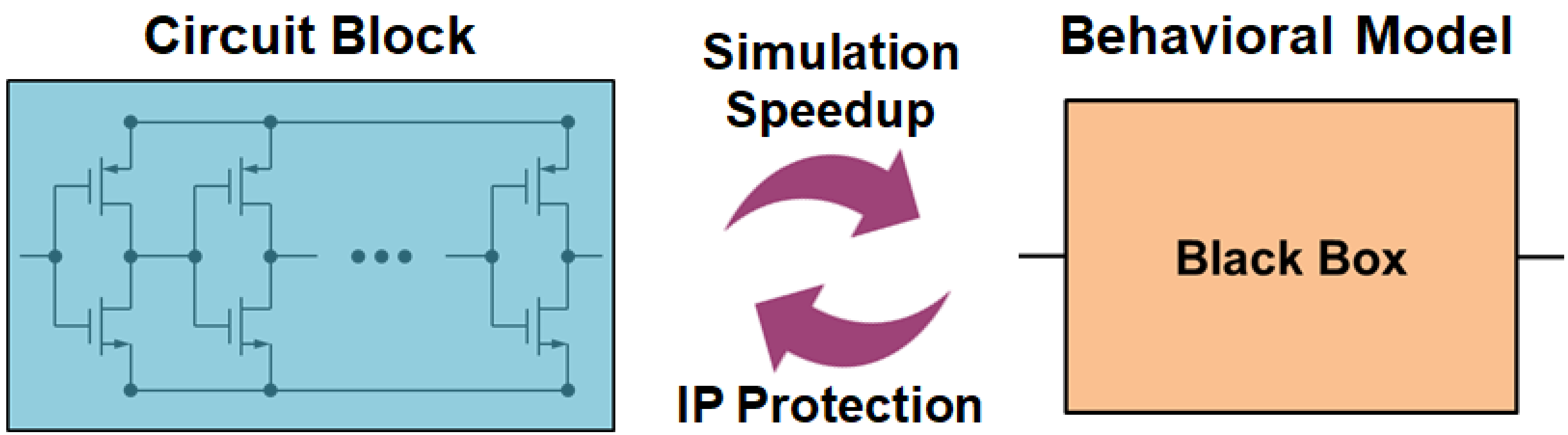
Faculty: Prof. Madhavan Swaminathan

Acknowledgements

This work was funded by the National Science Foundation under Grant No. CNS 16-24731 and the industry members of the CAEML IUCRC.

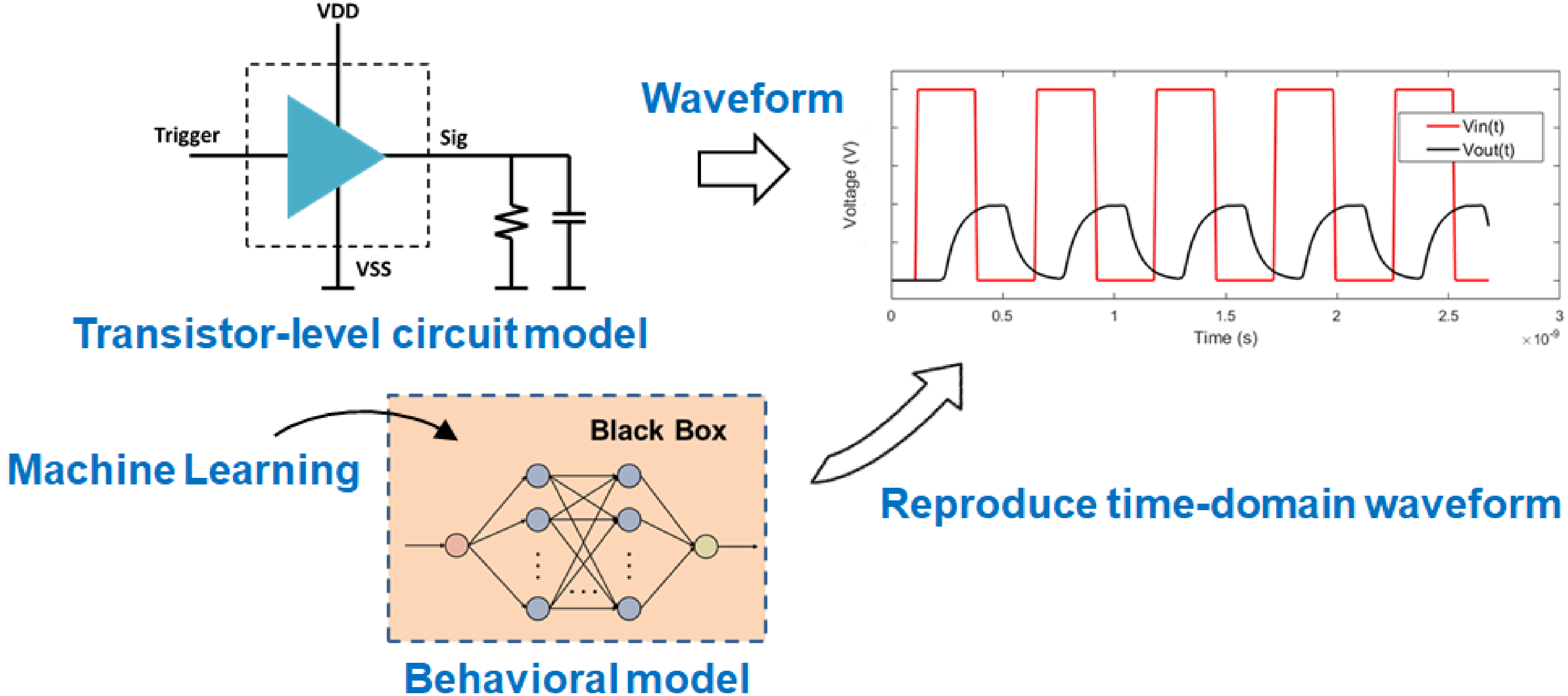
Industry Team: Qualcomm, Intel

Motivation



- Behavioral models achieve faster simulation compared to transistor-level models with certain level of accuracy
- Behavioral models treat transister-level circuits and systems as a black box, and thus protect intellectual property (IP)

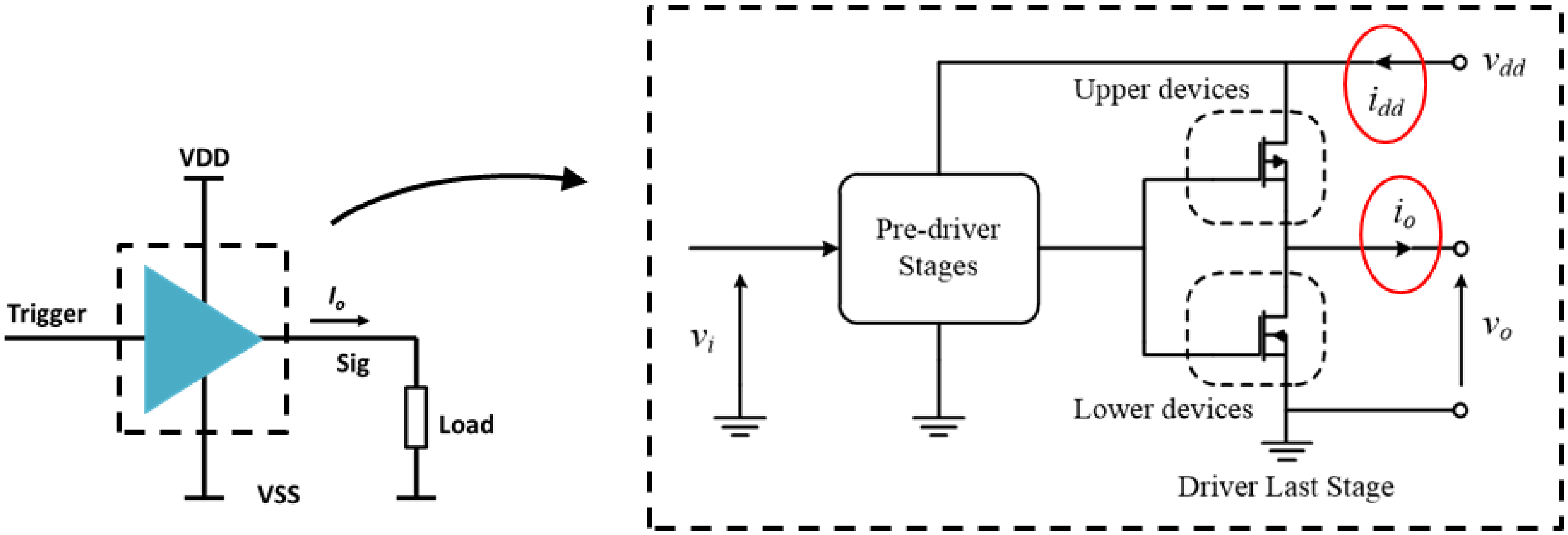
Behavioral Modeling Using Machine Learning



- **Create behavioral models that capture the nonlinear behavior of transister-level circuits using machine learning**
- **Behavioral models should be compatible with the existing commercial circuit simulators, e.g., Hspice, Spectre**

Cadence Spectre User Guide

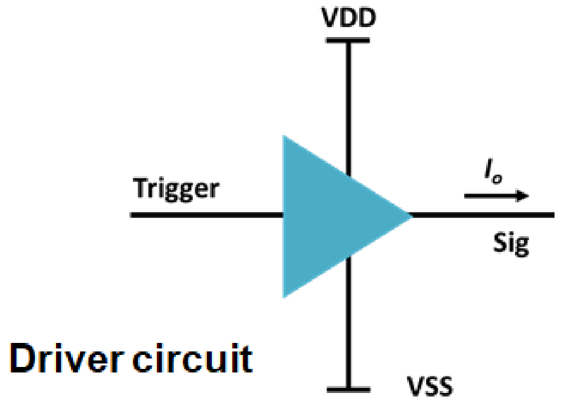
Driver Behavioral Modeling

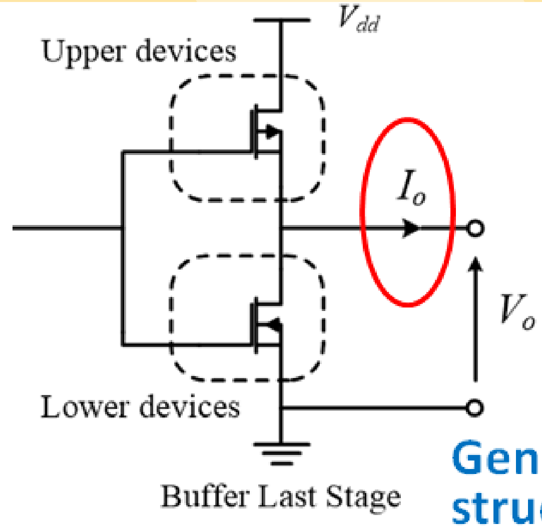


Driver circuit **Generic driver structure**

- ❑ Behavioral modeling is done by modeling the output port's current i_o and supply port's current i_{dd}

Model Structure





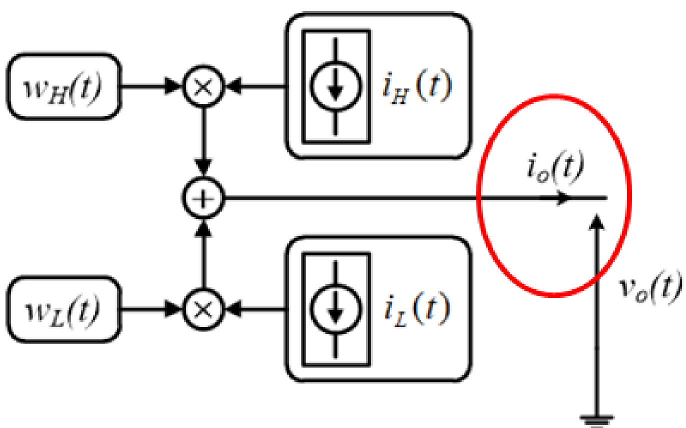
Generic structure

Output current:

Sub-models

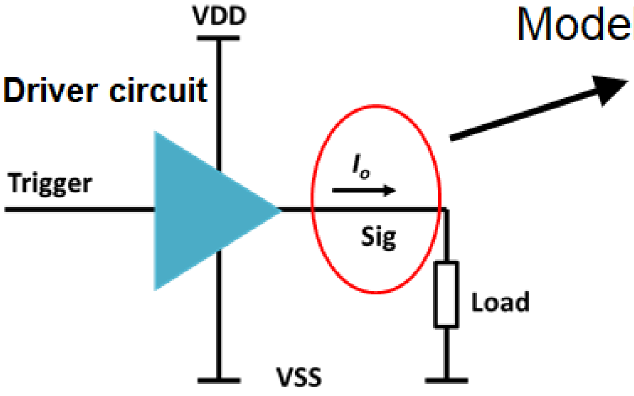
$$i_o(t) = w_H(t) \cdot i_H(t) + w_L(t) \cdot i_L(t)$$

Weighting functions



Behavioral Model

Output port



Model equation: $i_o(t) = w_{o,1}(t) \cdot i_{o,1}(t) + w_{o,2}(t) \cdot i_{o,2}(t)$

Modeling method:

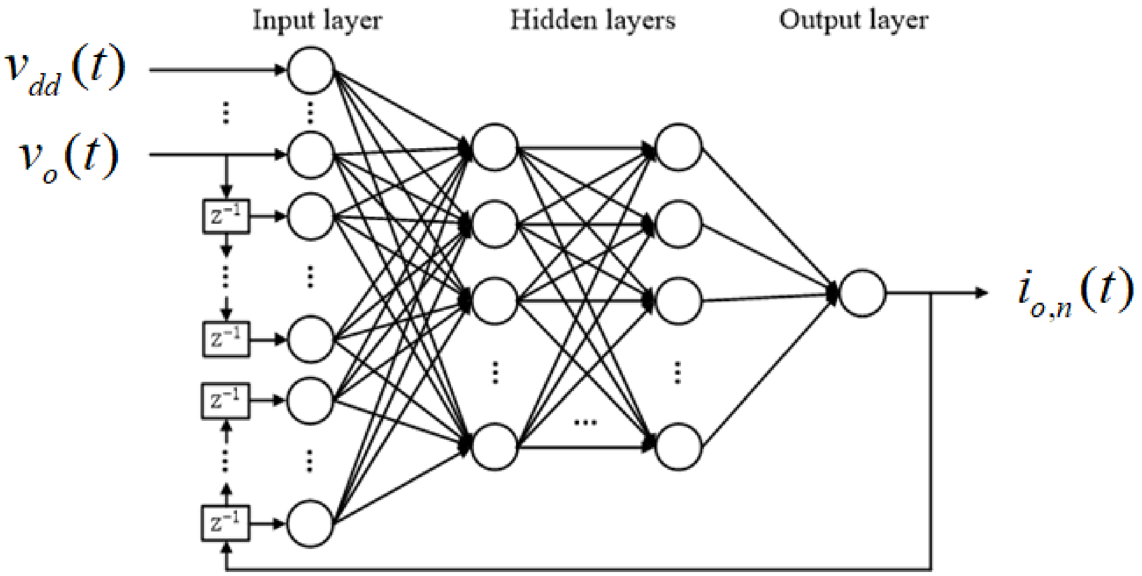
- Sub-models $i_{o,1}(t)$ and $i_{o,2}(t)$
  - Capture output port's nonlinear dynamic I-V relation
  - $i_{o,1}(t)$ for HIGH state
  - $i_{o,2}(t)$ for LOW state
- Weighting functions $w_{o,1}(t)$ and $w_{o,2}(t)$
  - Control state transitions according to the input signal

[1] Bhyrav Mutnury, PhD thesis, Georgia Tech

Learning Sub-models

- Sub-models are captured using recurrent neural networks (RNNs):

$$i_{o,n}(t) = f_n^{RNN}\begin{pmatrix} v_o(t), v_o(t-\Delta t), v_o(t-2\Delta t), \\ i_{o,n}(t-\Delta t), i_{o,n}(t-2\Delta t), \\ v_{dd}(t), v_{dd}(t-\Delta t), v_{dd}(t-2\Delta t) \end{pmatrix}; \quad n = 1,2$$

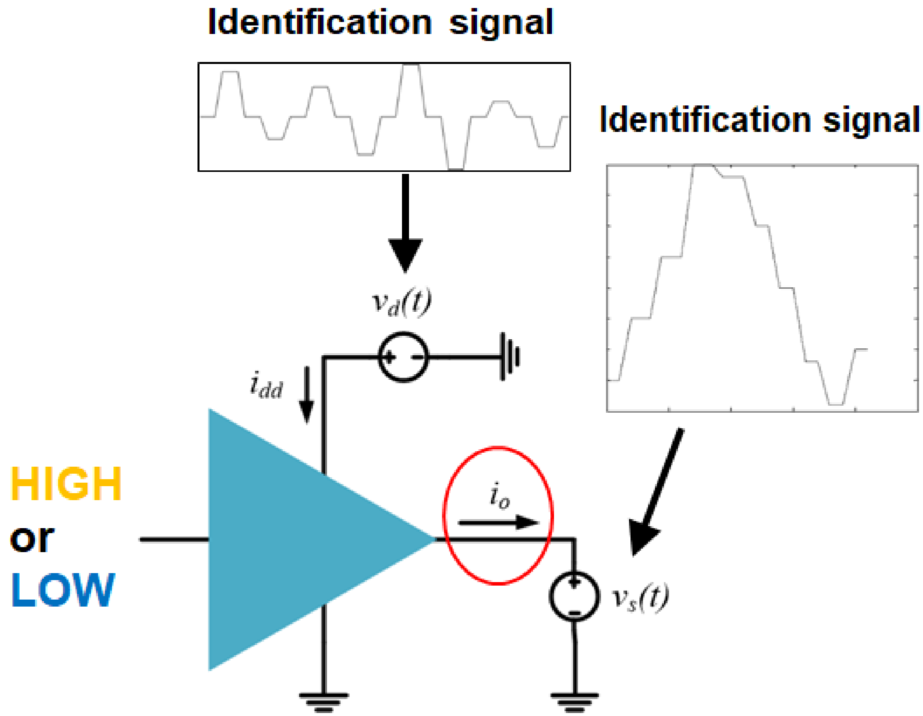


RNN structure

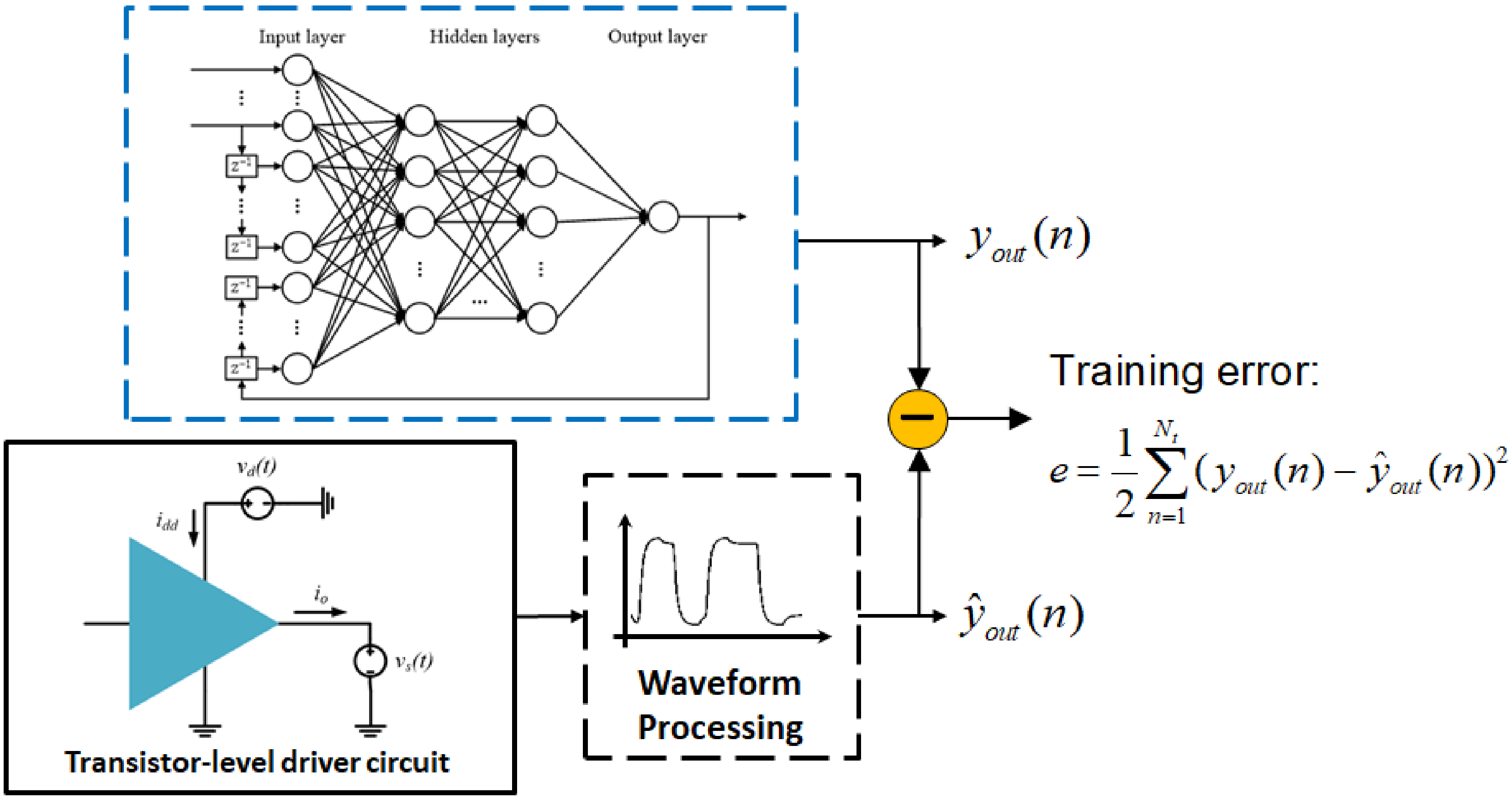
Capture Sub-models

Capturing sub-model $i_{o,1}$:
- Input fixed HIGH
- Apply voltage waveform to the output port and supply port (Identification signals)

Capturing sub-model $i_{o,2}$:
- Input fixed LOW
- Apply voltage waveform to the output port and supply port (Identification signals)



Sub-model Training

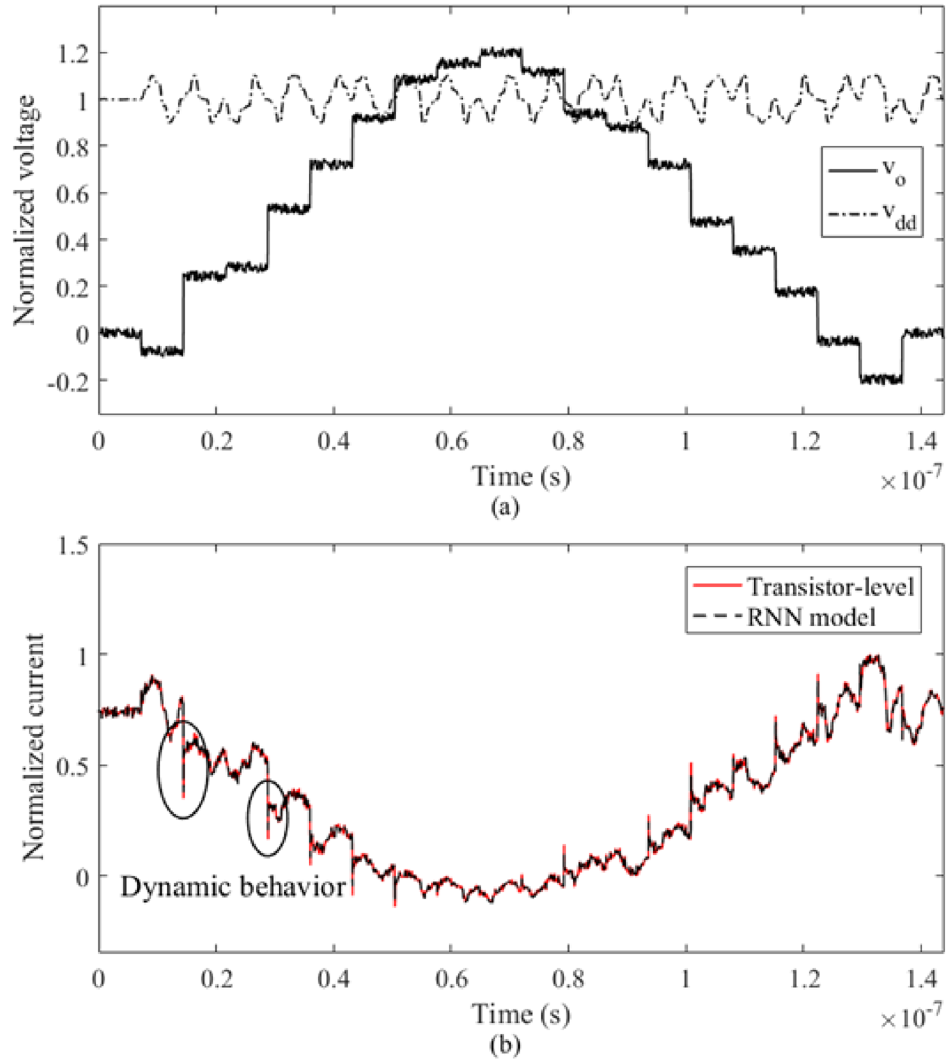


- ❑ Transistor-level driver circuit simulation waveform data is used for training
- ❑ The RNN sub-models are trained to minimize the training error

RNN Training Results

Identification signals:

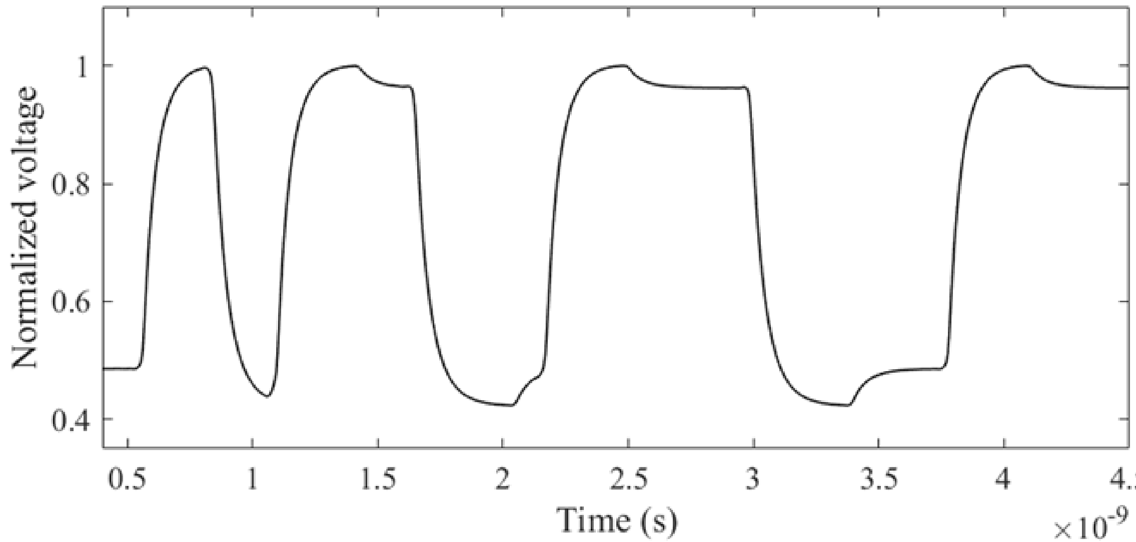
Output current:
(input HIGH)



Capturing Weighting Functions

$$i_o(t) = \boxed{w_{o,1}(t)} \cdot i_{o,1}(t) + \boxed{w_{o,2}(t)} \cdot i_{o,2}(t)$$

Output waveform with pre-emphasis

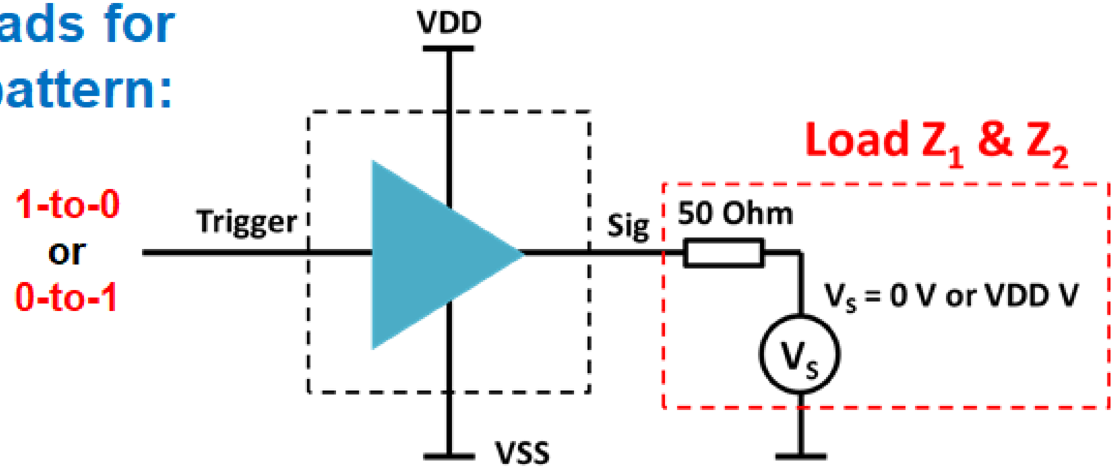


- ❑ Pre-emphasis driver has multiple output voltage levels
- ❑ The pre-emphasis effects are captured using weighting functions

Weighting Functions Extraction

$$i_o(t) = \boxed{w_{o,1}(t)} \cdot i_{o,1}(t) + \boxed{w_{o,2}(t)} \cdot i_{o,2}(t)$$

Two different loads for one switching pattern:

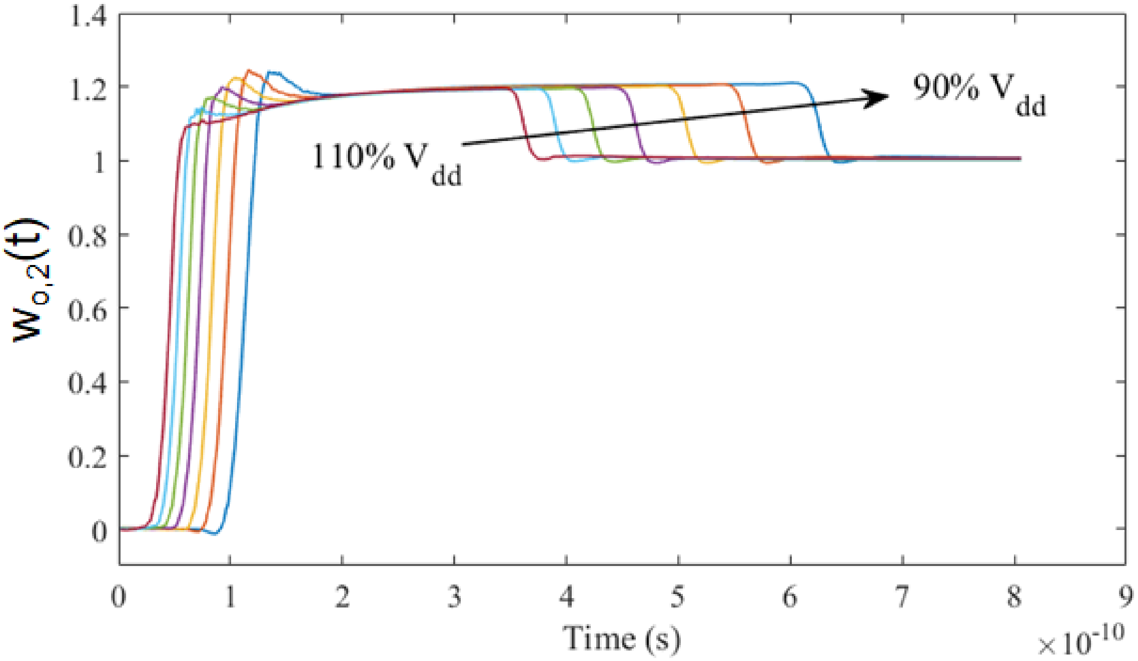


Extraction using matrix inversion:

$$\begin{bmatrix} w_{o,1}(t) \\ w_{o,2}(t) \end{bmatrix} = \begin{bmatrix} i_{o,1}^{Z_1}(t) & i_{o,2}^{Z_1}(t) \\ i_{o,1}^{Z_2}(t) & i_{o,2}^{Z_2}(t) \end{bmatrix}^{-1} \begin{bmatrix} i_o^{Z_1}(t) \\ i_o^{Z_2}(t) \end{bmatrix}$$

Influence of Vdd on Weighting Functions

Extracted $w_{o,2}(t)$ with V_{dd} variations:



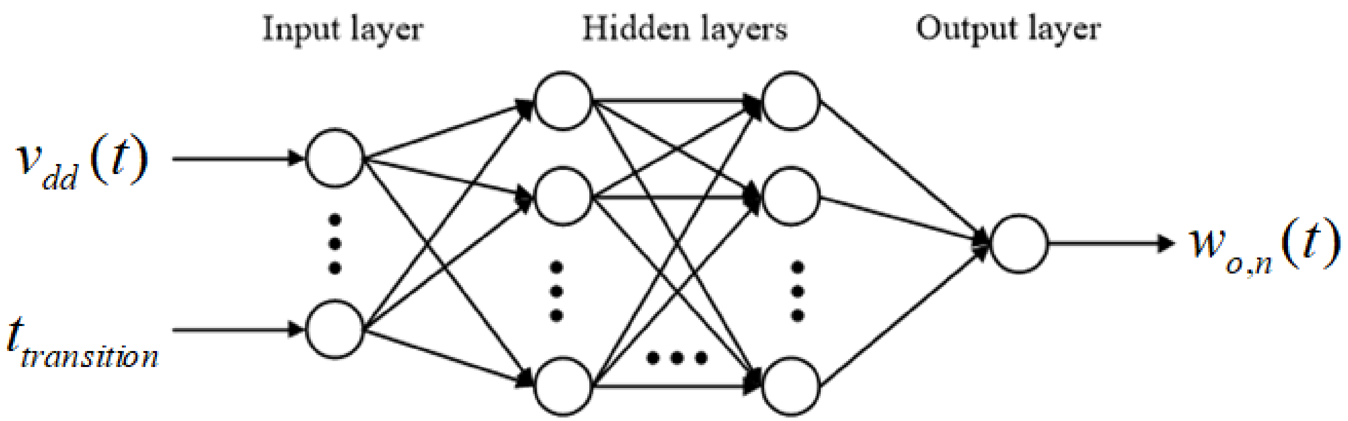
- The supply voltage V_{dd} has an influence on both the transition delay and the pre-emphasis behavior
- The weighting function extraction is performed for multiple V_{dd} values sampled within the dynamic operation range of the supply voltage

Learning Extracted Weighting Functions

- Weighting functions are captured using feed-forward neural networks (FFNNs):

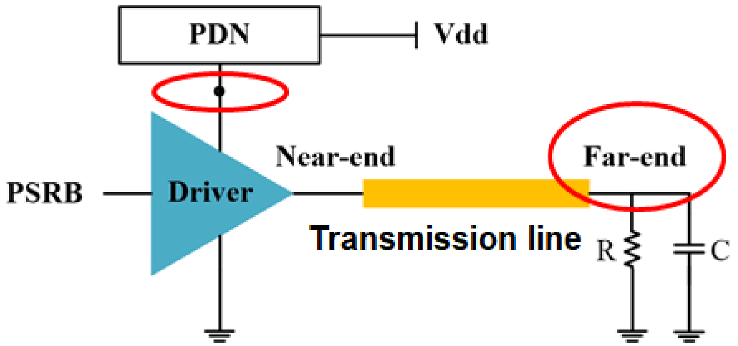
$$\begin{cases} w_{o,1}(t) = w_{o,1}^{FFNN}(v_{dd}, t_{transit}) \\ w_{o,2}(t) = w_{o,2}^{FFNN}(v_{dd}, t_{transit}) \end{cases}$$

$t_{transit}$: transition time

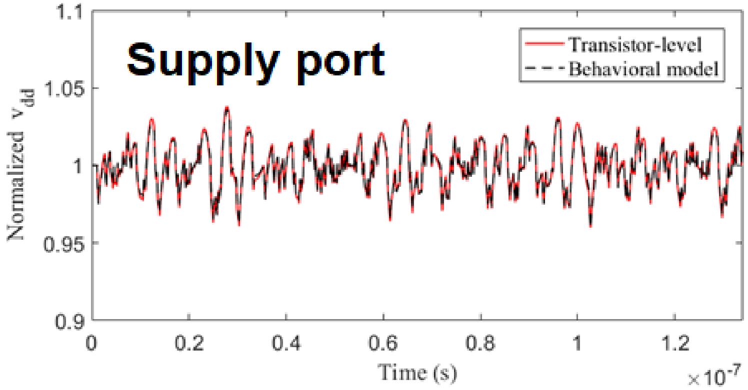


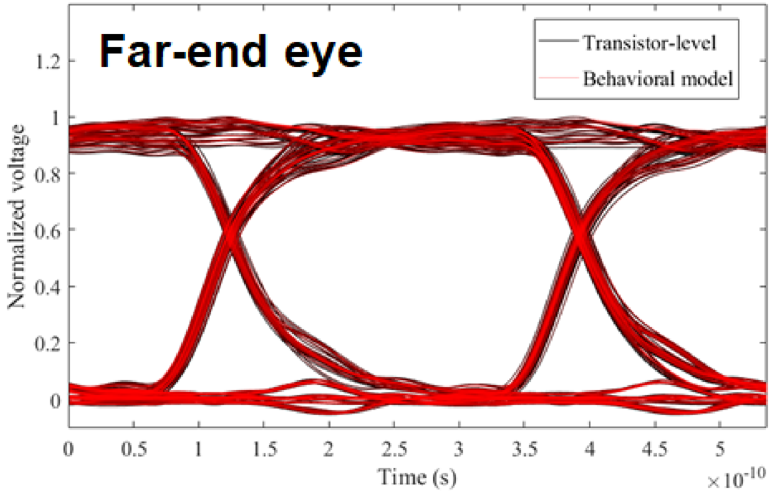
FFNN structure

Modeling Results



Simulator: Spectre





- Model achieves ~300X simulation speed-up

Summary

	Proposed model
Use of Machine Learning	√
Compatible with Verilog-A & Spice	√
Capturing the nonlinear dynamic behavior of the port current	√
Capturing the influence of V_{dd} variation on the transition and pre-emphasis behavior	√
Significant simulation speed-up (~300X)	√