The University of Aizu

Research Progress Seminar (RPS)

Spiking Neural Network with 3-D IC-based Stacking Memory

Ngo-Doanh NGUYEN
M5262108

2023-01-06

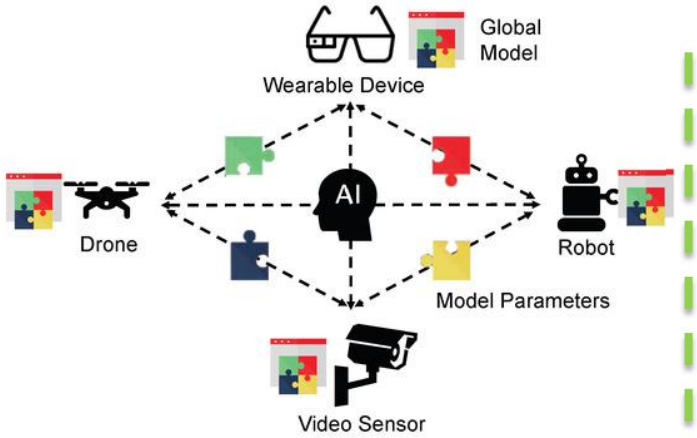
Agenda

1. Motivation & Background
2. Goals
3. Approach
4. Schedule
5. Reference

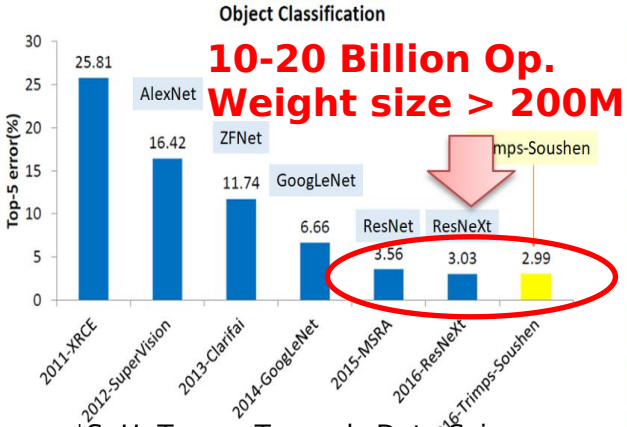
Agenda

1. Motivation & Background
2. Goals
3. Approach
4. Schedule
5. Reference

Motivation

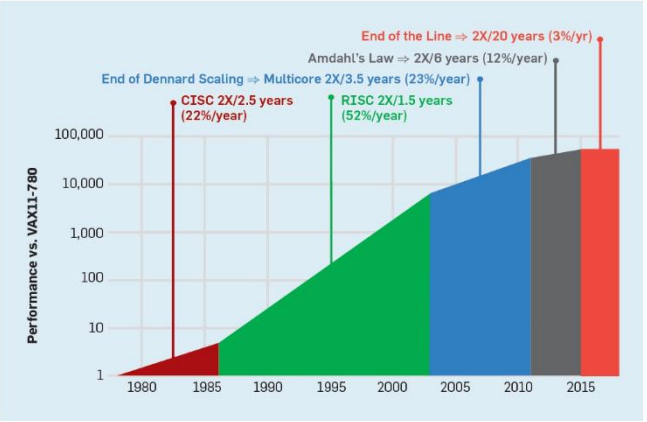


*Y. Du. Decentralized Smart IoT. Encyclopedia



10-20 Billion Op.
Weight size > 200MB

*S. H. Tsang. Towards Data Science



*J.Hennessy, D. Patterson 2019 CACM

Computational Power for Edge Devices

AI Enabled Devices

- *Improve Data Transfer Efficiency*
  - *+ Reduce Latency*
  - *+ Reduce Power*

High Complexity for Edge Devices

Spiking Neural Net.

- *Reduce Power Consumption*
- *Reduce Memory Footprint*
- *Reduce Hardware Area*

End of Moore's Law

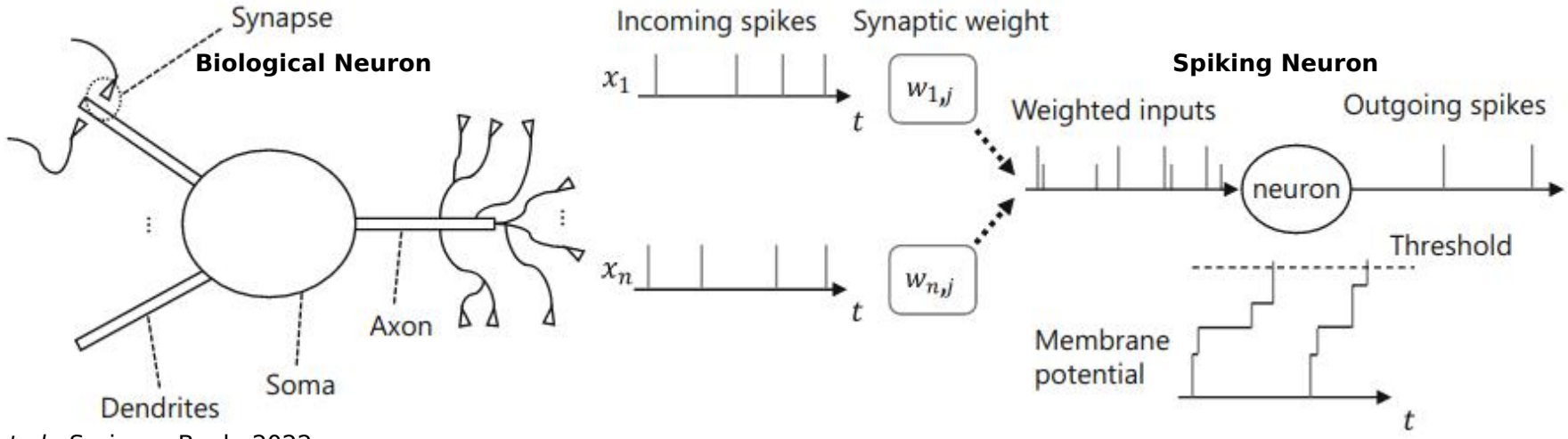
3-D Stacking Arch.

- *Reduce Latency*
- *Reduce Power Consumption*
- *Reduce Hardware Area*

=> Spiking Neural Network with 3-D IC-based Stacking Memory!!!

Spiking Neural Network

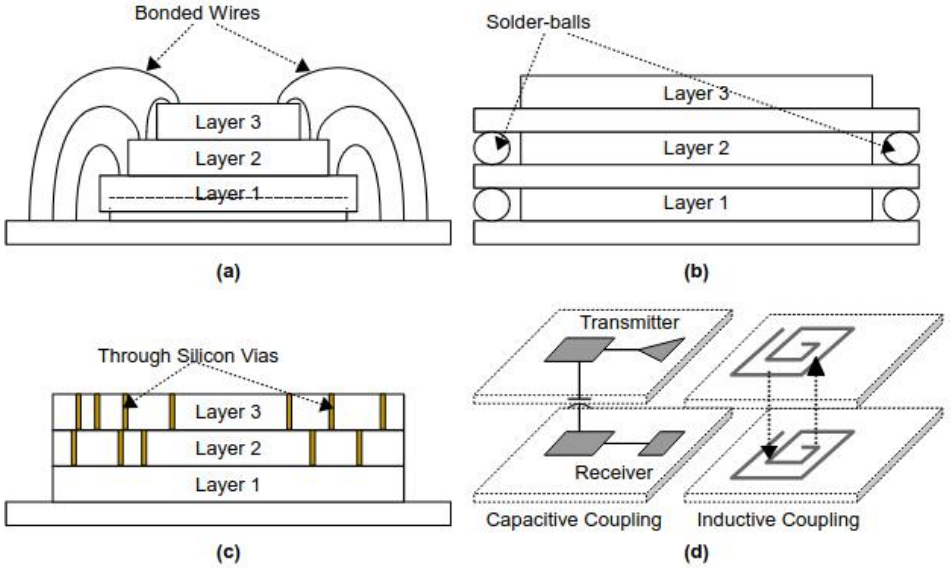
- SNNs are artifical neural network mimiced biological brian.
- There are three major parameters
  - Incoming spikes
  - Synaptic weights
  - Neuron's internal parameters (membrane potential, threshold, ...)



*A. Ben *et al.*, Springer Book, 2022

3-D IC Opportunity

- What is 3-D IC?



(a) Wire bonding; (b) Solder balls; (c) Through Silicon Vias; (d) Wireless stacking

- Stacking multiple layers to obtain smaller footprint & shorter interconnect & power.

- Why 3-D IC?

#Bit	Kogge-Stone Adder		Log Shifter 16		Log Shifter 32	
	16-bit				32-bit	
	Delay	Power	Delay	Power	Delay	Power
2 planes	-20.2%	-8%	-13.4%	-6.5%	-28.4%	-8%
3 planes	-23.6%	-15%	-	-	-	-
4 planes	-32.7%	-22%	-	-	-	-

Performance & Power: 3D vs 2D architecture*

- #Stacking Layers ↑ ⇔ Power & Performance & Area (PPA) ↑

=> 3-D Stacking Memory for SNN is promissing

*B. Valdyanathan *et al.*, *ICVLSI Design*, 2007

3-D IC SNN

- SNN is a light weight algorithm
  - Low memory footprint
  - Low power consumption
- 3-D technology leverages the performance of ICs
  - Reduce hardware footprint
  - Reduce power consumption
  - Increase performance

=> 3-D IC SNN achieves ultra-low-power & high-performance for edge devices' applications.

Agenda

1. Motivations & Background
2. Goals
3. Approach
4. Schedule
5. Reference

Goals

- Design a hardware architecture of SNN with 3-D memory stacking
- Purpose a strategy to reduce power consumption

Agenda

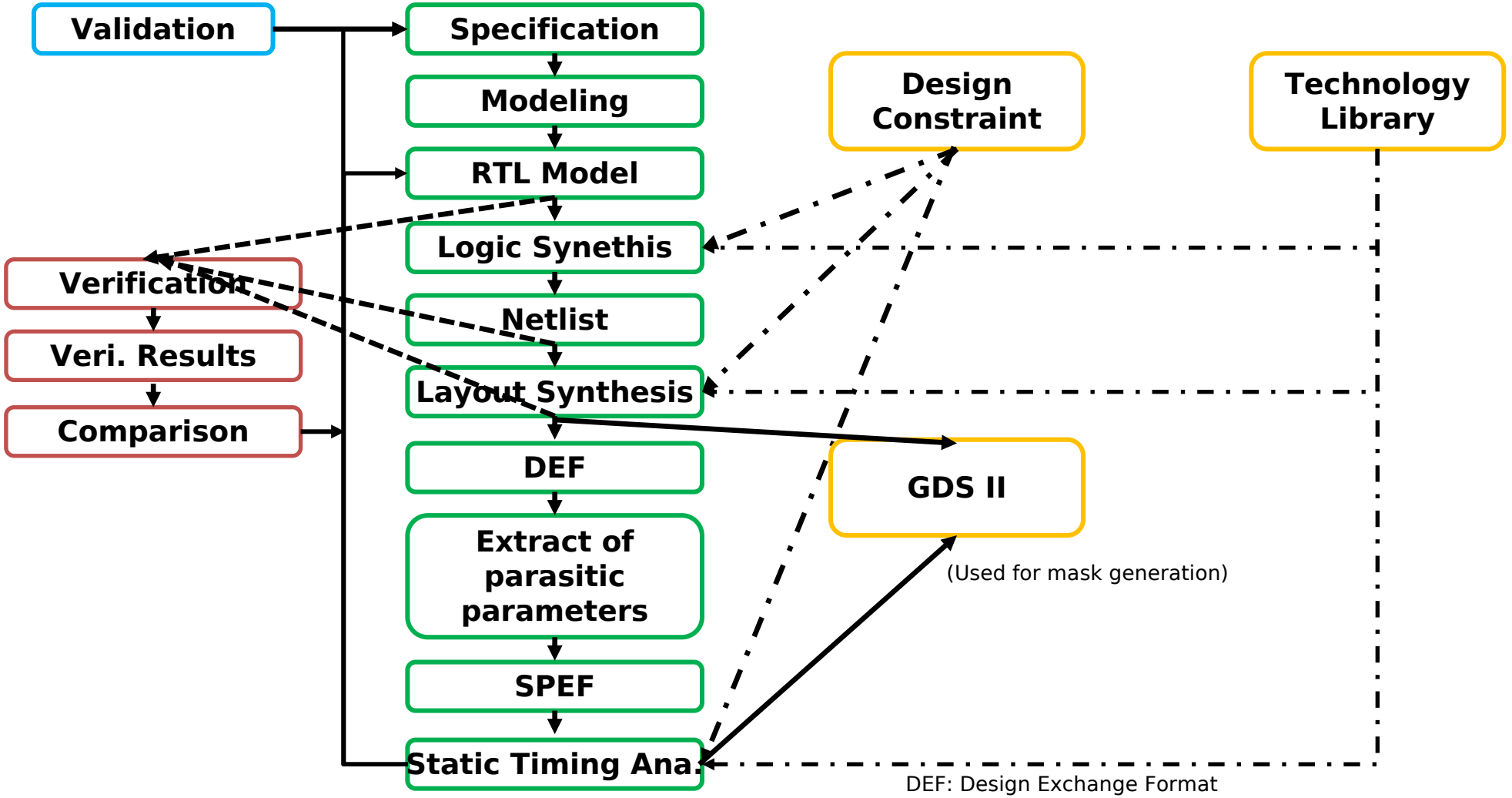
1. Motivations & Background
2. Goals
3. Approach
4. Schedule
5. Reference

Approach

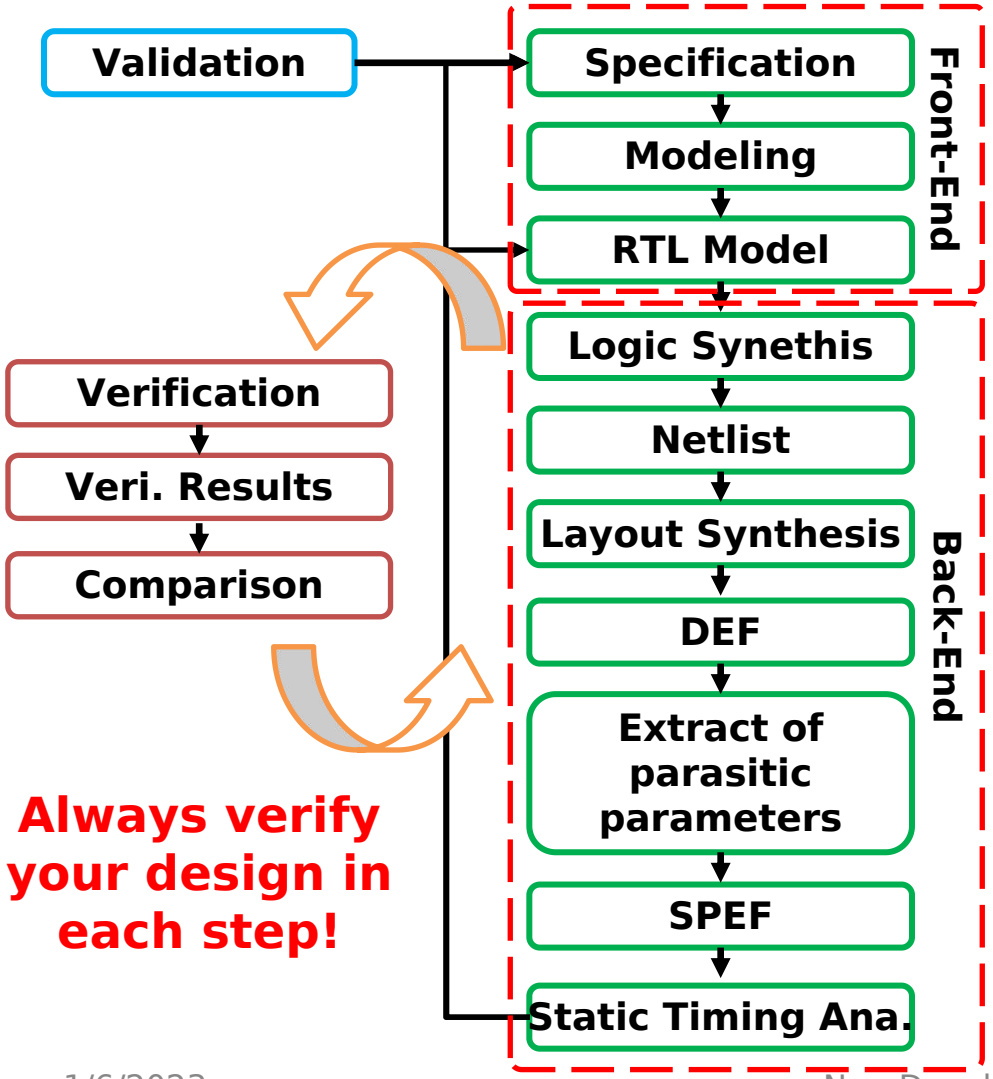
- Design a hardware architecture of SNN with 3-D memory stacking
- Purpose a strategy to reduce power consumption

=> Need to understand EDA tools.

Hardware Design Flow (1)

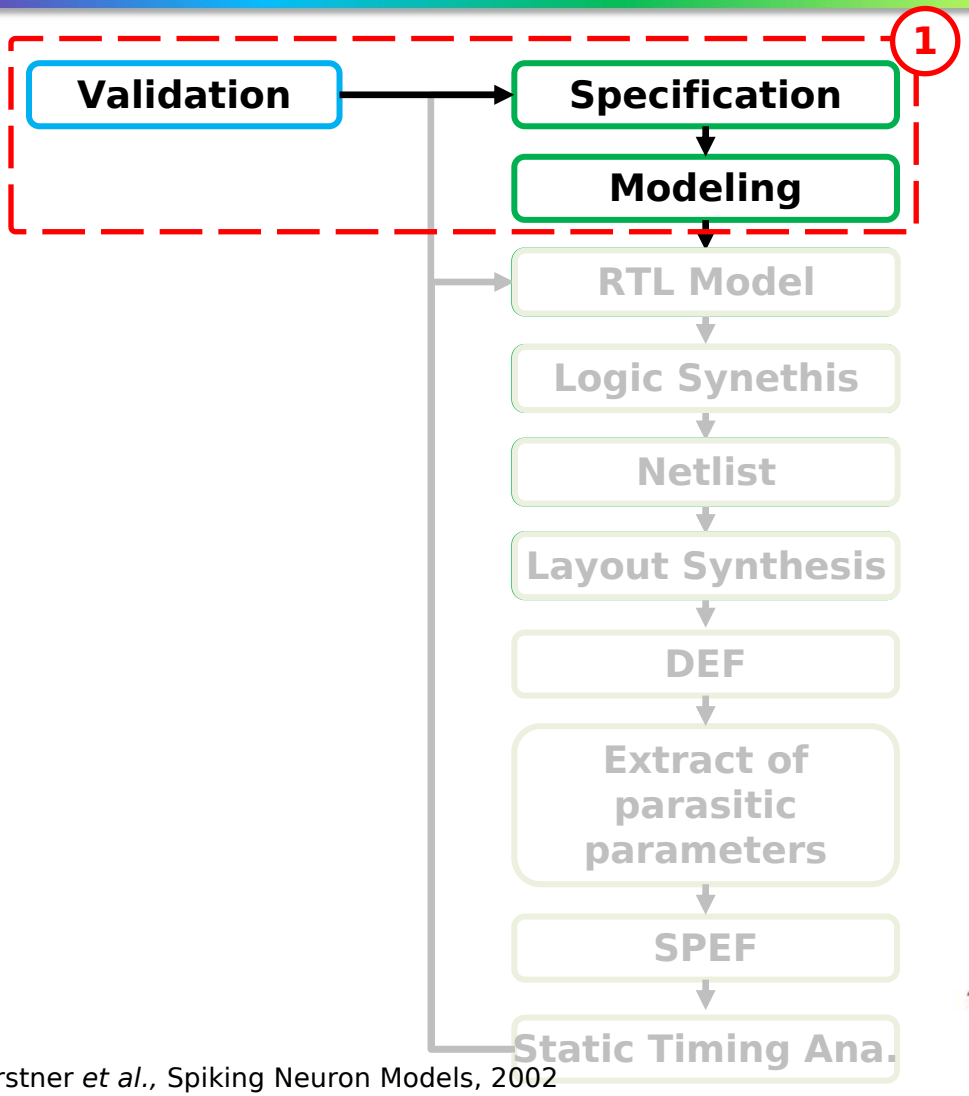


Hardware Design Flow (2)



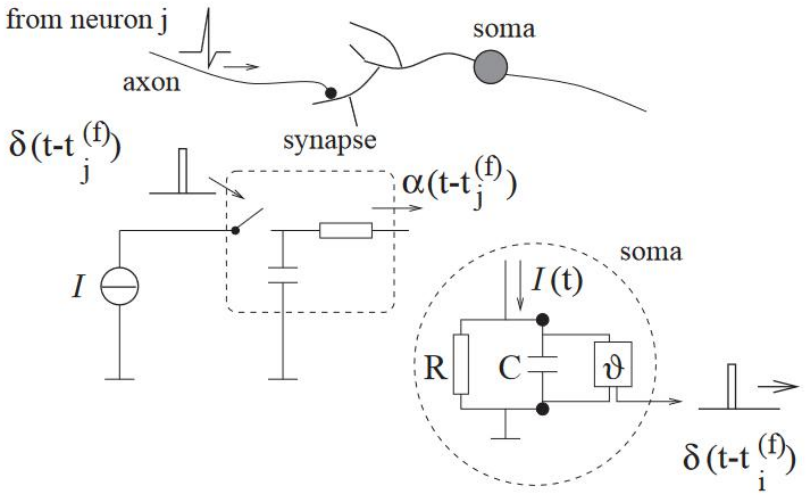
- 4 basic steps:
  1. Define hardware configurations
  2. Design (& verify) hardware's behavior with Hardware Description Language *(HDL)*
  3. Synthesis (& verify) hardware design into netlist *(logic-gate level)*
  4. Layout (& verify) hardware design into solid netlist *(transitor level)*
- Step 1 & 2 is called **RTL Design** or **Front-End**.
- Step 3 & 4 is called **Physical Design** or **Back-End**.

Hardware Configurations



1. Define hardware configurations

From the integrate-and-fire behavior of neuron to schematic circuit*.

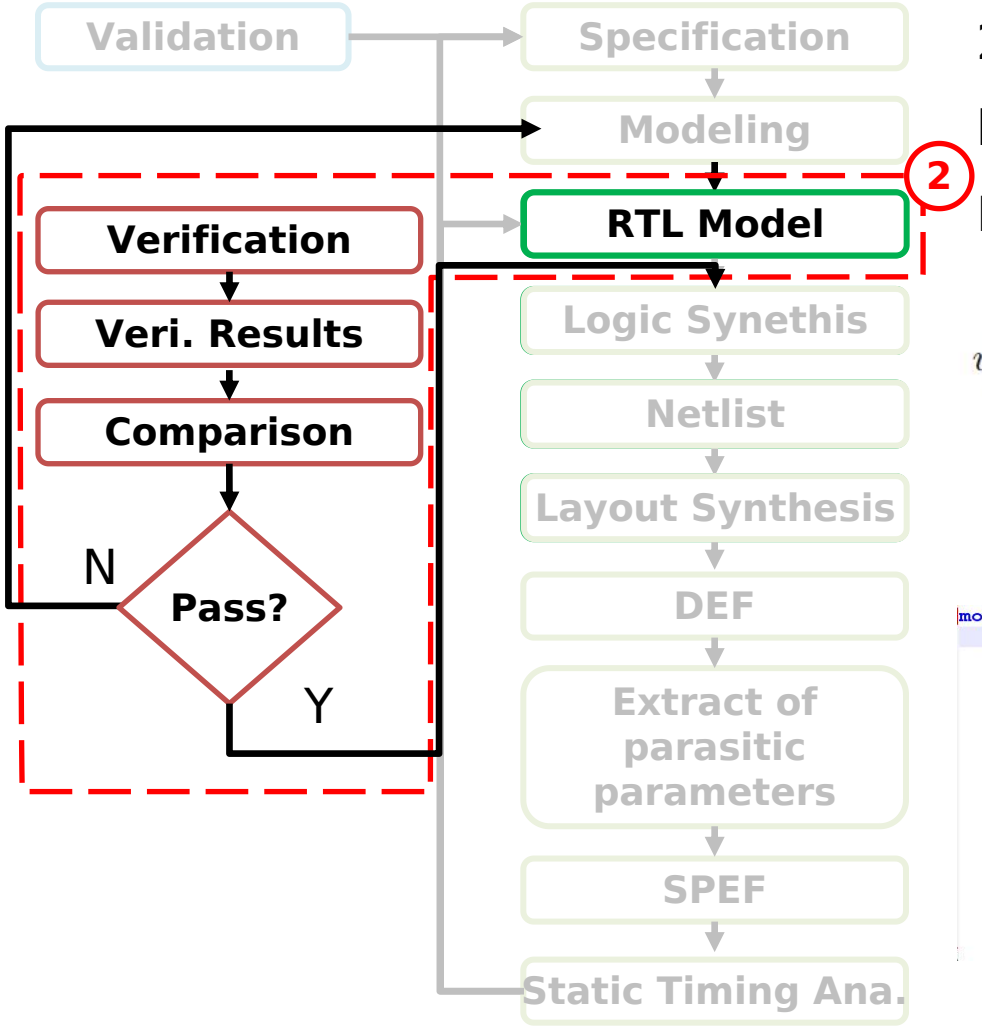


With the support of mathematical model (Leaky Integrate-and-Fire)

$$u(t) = u_r \exp\left(-\frac{t-\hat{t}}{\tau_m}\right) + \frac{1}{C}\int_0^{t-\hat{t}} \exp\left(-\frac{s}{\tau_m}\right) I(t-s)\,\mathrm{d}s\,.$$

*W. Gerstner *et al.*, Spiking Neuron Models, 2002

RTL Design



2. Design (& verify) hardware's behavior with Hardware Description Language *(HDL)*.

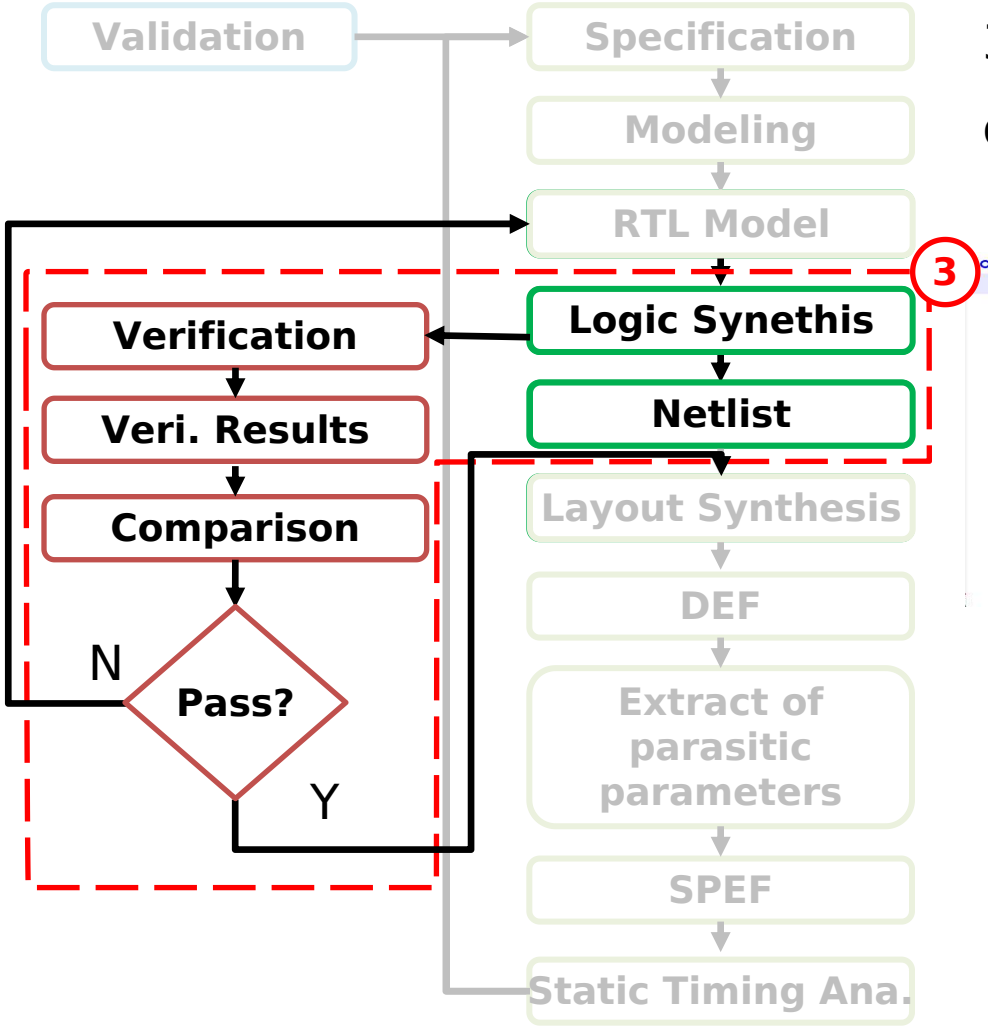
Leaky Integrate-and-Fire

$$u(t) = u_r \exp\left(-\frac{t-\hat{t}}{\tau_m}\right) + \frac{1}{C}\int_0^{t-\hat{t}} \exp\left(-\frac{s}{\tau_m}\right) I(t-s)\,\mathrm{d}s\,.$$

RTL Code

```
module SNPC (
    input                                   clk,            // system clock
    input                                   rst_n,          // system reset
    input [WEIGHT_ADDR_WIDTH-1:0]           address_1,
    input [3:0]                             i_ram_ena,
    input                                   cs_1,
    input                                   we_1,
    input                                   oe_1,
    inout [NEURAL_ARRAY_WIDTH*(WEIGHT_WIDTH)-1:0]                          data_1,
    output [SPK_ARRAY_WIDTH*NEURAL_ARRAY_WIDTH*WEIGHT_WIDTH-1:0]           o_weight,
    input                                   i_start,
    input [SPK_ARRAY_WIDTH-1:0]             i_spk_arry,     // input spike with weight
    input                                   i_svalid,       // valid of spike
    output [NEURAL_ARRAY_WIDTH*(LIF_NEURON_OVRFL_WIDTH+WEIGHT_WIDTH)-1:0]  o_V,
    output [NEURAL_ARRAY_WIDTH-1:0] o_spike,
    output                                  o_svalid,
  );
```

Hardware Synthesis



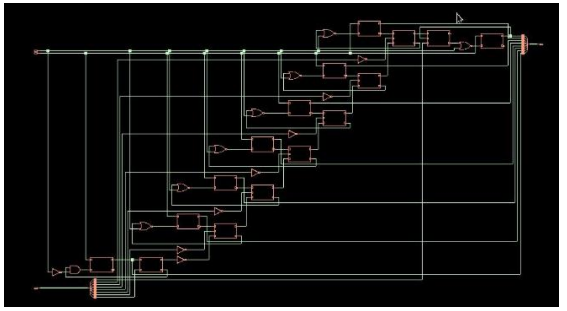
3. Synthesis (& verify) hardware design into netlist *(logic-gate level).*

RTL Code

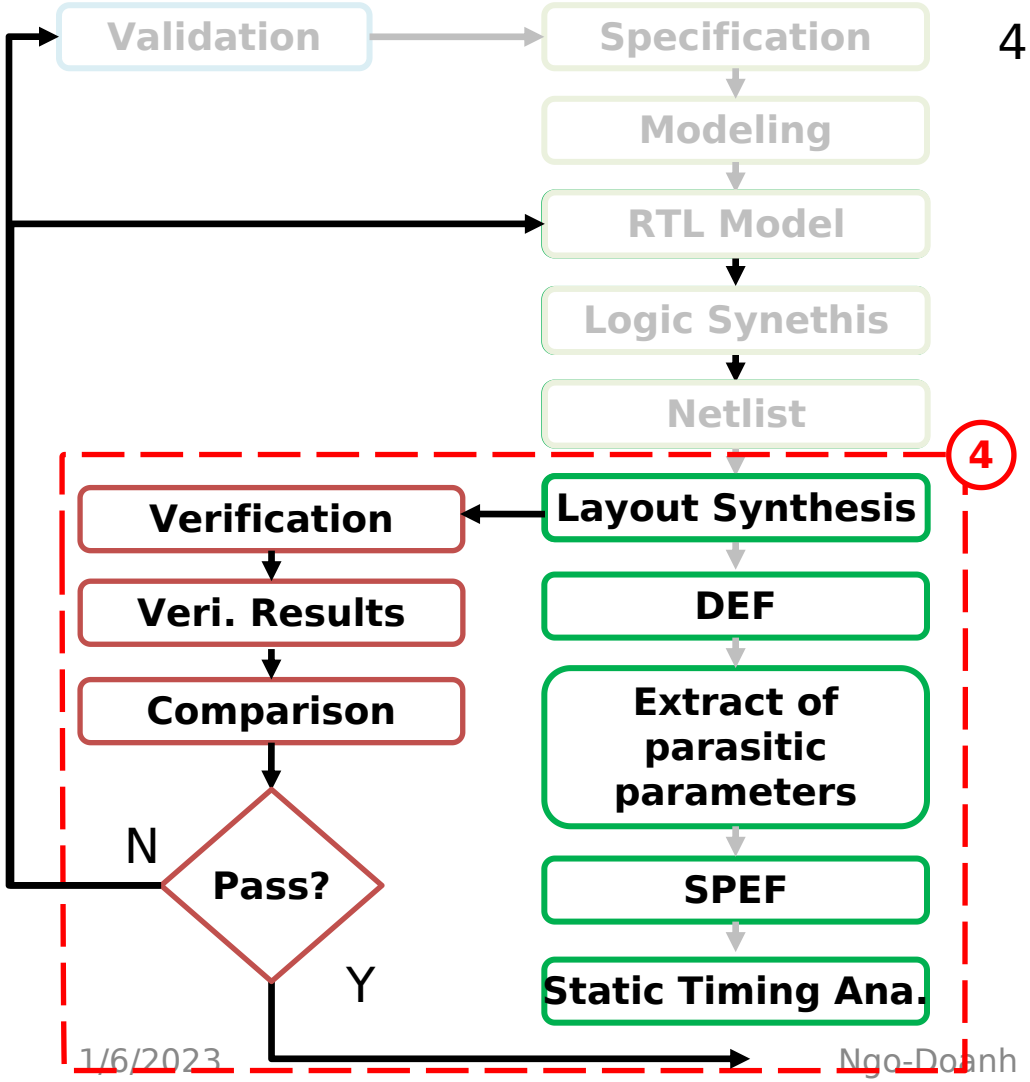
```
odule SNPC (
    input                                clk,            // system clock
    input                                rst_n,          // system reset
    input [WEIGHT_ADDR_WIDTH-1:0]        address_1,
    input [3:0]                          i_ram_ena,
    input                                cs_1,
    input                                we_1,
    input                                oe_1,
    inout [NEURAL_ARRAY_WIDTH*(WEIGHT_WIDTH)-1:0]                          data_1,
    output [SPK_ARRAY_WIDTH*NEURAL_ARRAY_WIDTH*WEIGHT_WIDTH-1:0]           o_weight,
    input                                i_start,
    input [SPK_ARRAY_WIDTH-1:0]          rry,      // input spike with weight
    input                                d,        // valid of spike
    output [NEURAL_ARRAY_WIDTH*(LIF      OVRFL_WIDTH+WEIGHT_WIDTH)-1:0]    o_V,
    output [NEURAL_ARRAY_WIDTH-1
    output                               d,
);
```

Netlist View

(logic-gate level)

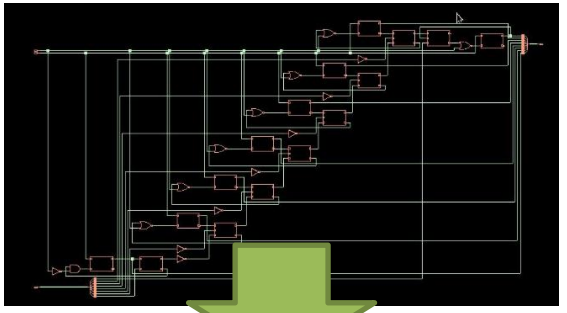


Hardware Layout

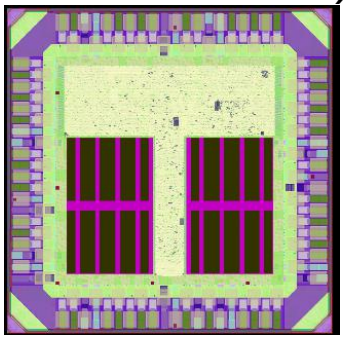


4. Layout (& verify) hardware design into solid netlist *(transitor level)*

Netlist View
(logic-gate level)



Layout View
(transitor level)



Cadence CAD

- Use Cadence Compute-Aided Design (CAD) tools for implementing hardware design flow
- Cadence Genus for logic synthesis
- Cadence Innovus for layout implementation

Implementation Results (1)

- Hardware results extracted from Genus

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 18.13-s027_1
  Generated on:           Nov 14 2022  04:26:50 pm
  Module:                 SNPC
  Technology libraries:   NangateOpenCellLibrary revision 1.0
                          sram_w2_b256_freepdk45_TT_1p0V_25C_lib
  Operating conditions:   typical (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                          Leakage    Dynamic      Total
    Instance      Cells   Power(nW)  Power(nW)   Power(nW)
------------------------------------------------------------
SNPC                3260 141543.468 3997657.697 4139201.165
  XBAR0             2273 104274.266 3440403.824 3544678.091
  LIF_GEN[8].LIF0     97   3692.299   50419.880   54112.178
  LIF_GEN[5].LIF0     97   3683.869   48352.252   52036.121
  LIF_GEN[3].LIF0     97   3679.472   47469.579   51149.051
  LIF_GEN[0].LIF0     97   3678.487   48202.825   51881.313
  LIF_GEN[6].LIF0     97   3667.318   50351.800   54019.118
  LIF_GEN[4].LIF0     97   3666.899   47034.805   50701.704
  LIF_GEN[9].LIF0     97   3662.349   44264.480   47926.829
  LIF_GEN[1].LIF0     97   3661.889   45698.487   49360.376
  LIF_GEN[7].LIF0     97   3656.398   46361.068   50017.466
  LIF_GEN[2].LIF0     97   3645.771   47552.836   51198.607
  CNTL0               15    542.412    8540.419    9082.831
```

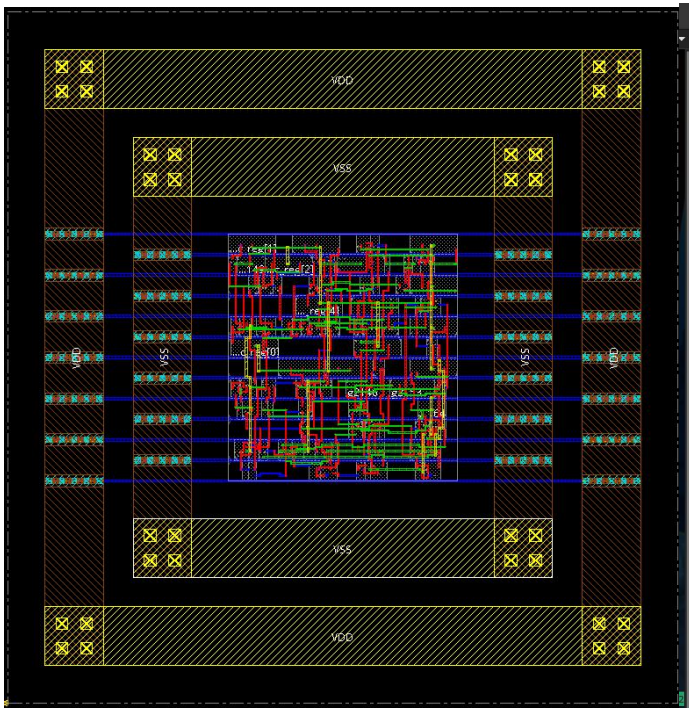
LIF Power Consumption

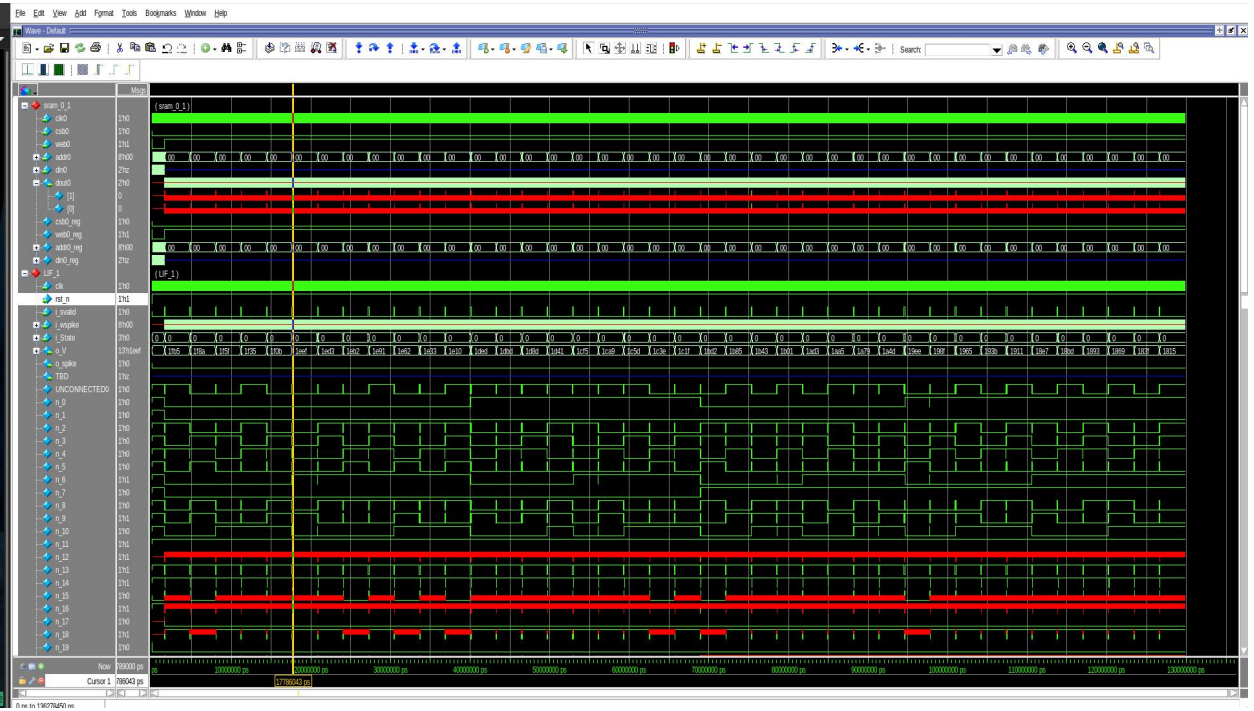
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 18.13-s027_1
  Generated on:           Nov 14 2022  04:26:51 pm
  Module:                 SNPC
  Technology libraries:   NangateOpenCellLibrary revision 1.0
                          sram_w2_b256_freepdk45_TT_1p0V_25C_lib
  Operating conditions:   typical (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

    Instance                          Module                          Cell Count  Cell Area  Net Area   Total Area     Wireload
-------------------------------------------------------------------------------------------------------------------------------
SNPC                                                                      3260 124230.313     0.000   124230.313 5K_hvratio_1_1 (D)
  XBAR0             xbar_LAYER_INDX0_WEIGHT_WIDTH8_SPK_ARRAY_WIDTH256_    2273 122495.195     0.000   122495.195 5K_hvratio_1_1 (D)
  LIF_GEN[9].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_26                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[8].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_27                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[7].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_28                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[6].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_29                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[5].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_30                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[4].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_31                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[3].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_32                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[2].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_33                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[1].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0_34                   97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  LIF_GEN[0].LIF0 LIF_neuron_WEIGHT_WIDTH8_OUTPUT_REG0                      97    170.772     0.000      170.772 5K_hvratio_1_1 (D)
  CNTL0             SNPC_cntrl                                              15     25.802     0.000       25.802 5K_hvratio_1_1 (D)

(D) = wireload is default in technology library
```

LIF Hardware Area

Implementation Results (2)

- Hardware results extracted from Innovus





LIF Layout **LIF's behavior simulation**

Agenda

1. Motivation
2. Goal
3. Approach
4. Schedule
5. Reference

Road Map

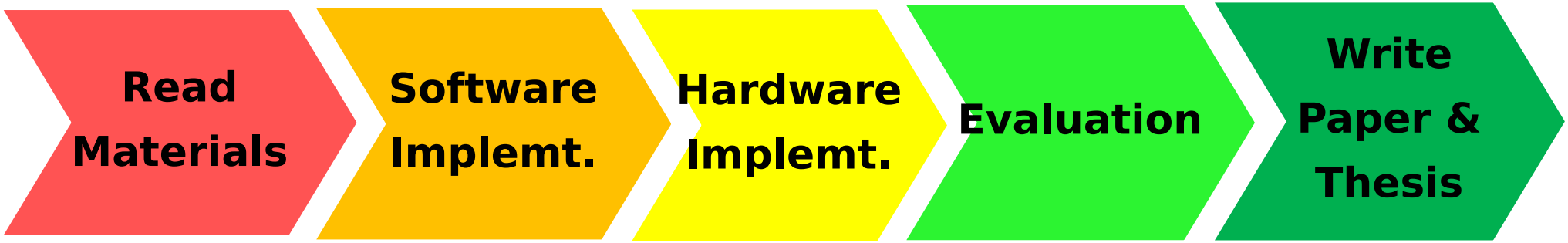
- Master's Course Schedule

1/23 - 5/23

- Build SW model as a golden reference for HW

12/23 - 3/24

- Evaluation power consumption & accuracy



10/22 - 1/23

- Get familiar with SNN

5/23 - 12/23

- Build HW model
- Optimize HW model

3/24 - 9/24

- Submit papers
- Complete thesis

Agenda

1. Motivation
2. Goal
3. Approach
4. Schedule
5. Reference

Reference

1. Du, Y. Decentralized Smart IoT. Encyclopedia. Available online: https://encyclopedia.pub/entry/8977 (accessed on 22 November 2022).
2. S. H. Tsang, https://towardsdatascience.com/review-refinenet-multi-path-refinement-network-semantic-segmentation-5763d9da47c1
3. John L. Hennessy, David A. Patterson, A New Golden Age for Computer Architecture, Communications of the ACM, February 2019, Vol. 62 No. 2, Pages 48-60, 10.1145/3282307
4. A. Ben, K. N, Dang, Neuromorphic Computing Principles and Organization, Springer, 2022
5. F. Akopyan et al., "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 10, pp. 1537-1557, Oct. 2015, doi: 10.1109/TCAD.2015.2474396.
6. M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," in IEEE Micro, vol. 38, no. 1, pp. 82-99, January/February 2018, doi: 10.1109/MM.2018.112130359.
7. B. V. Benjamin et al., "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," in Proceedings of the IEEE, vol. 102, no. 5, pp. 699-716, May 2014, doi: 10.1109/JPROC.2014.2313565.
8. B. Vaidyanathan, W.-L. Hung, F. Wang, Y. Xie, V. Narayanan, and M. J. Irwin, “Architecting microprocessor components in 3d design space,” in 20th International Conference on VLSI Design, 2007. Held Jointly with 6th International Conference on Embedded Systems., pp. 103–108, IEEE, 2007.

The University of Aizu

Thank you

for your attention.