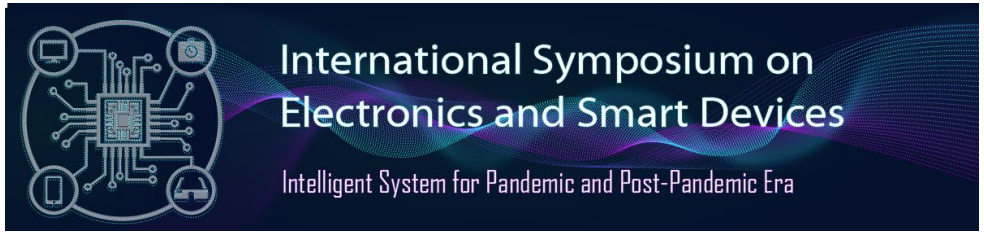
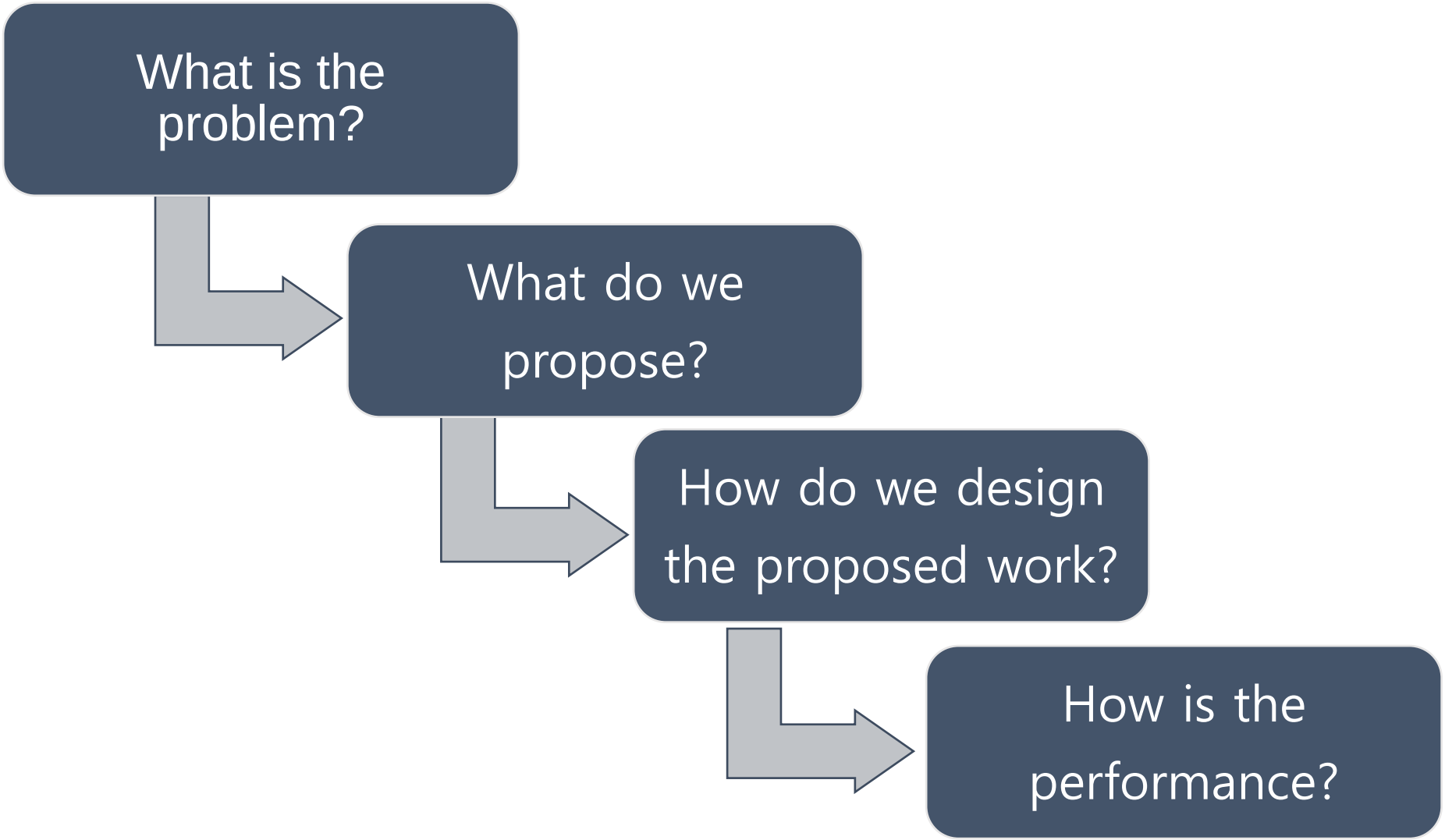


Architecture Design of Q-Learning Accelerator for Intelligent Traffic Light Control System

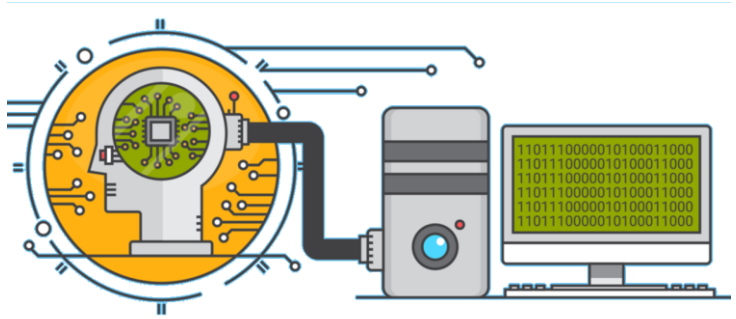
Nana Sutisna, Zulfikar N. Arifuzzaki, Infall Syafalni,
Rahmat Mulyawan, Trio Adiono

2022 International Symposium on Electronics and Smart Devices
November 8-9, 2022, Hybrid Conference, Bandung

Outline

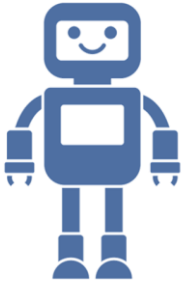

What is the problem?
What do we propose?
How do we design the proposed work?
How is the performance?

What is the problem?



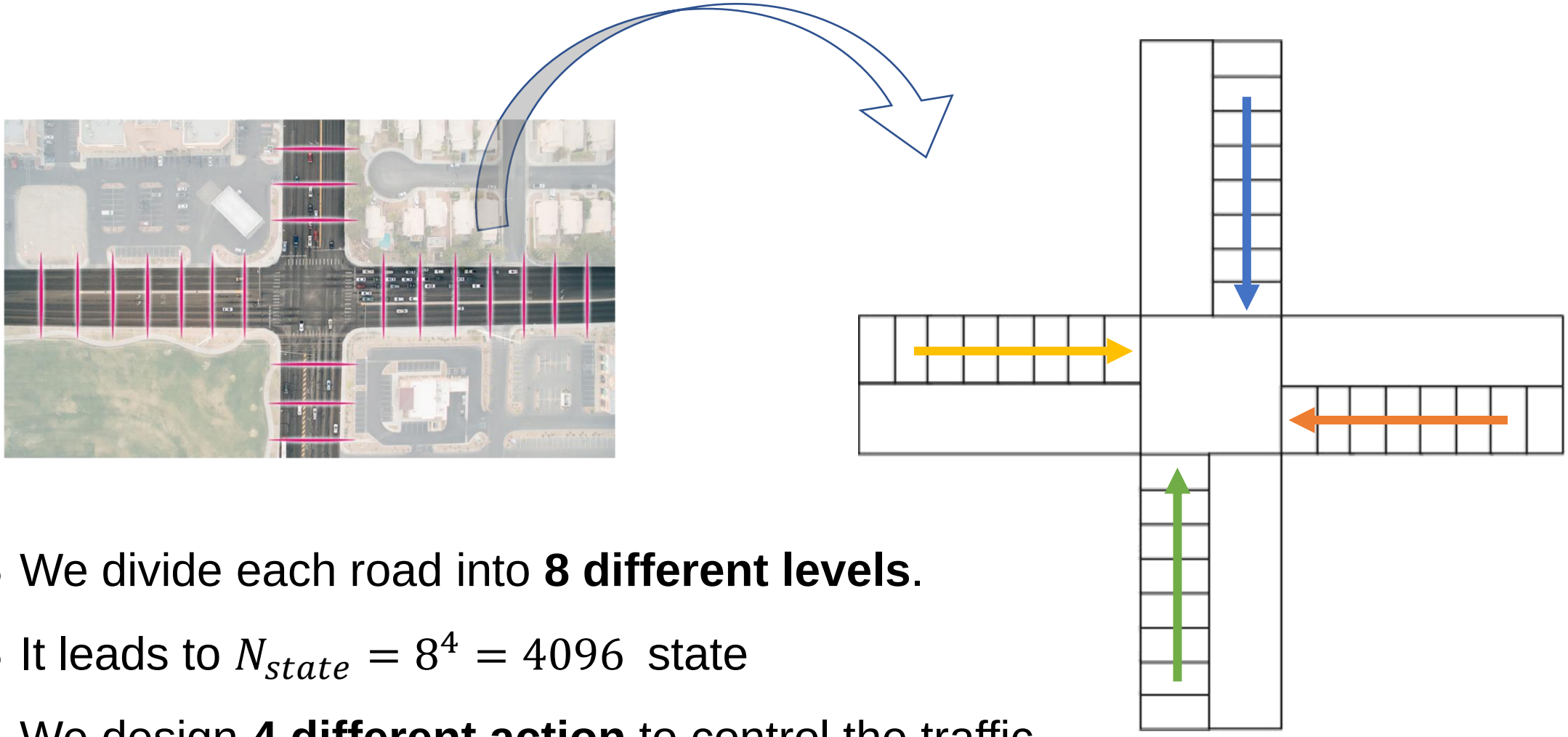
The development on machine learning to solve everyday-life problems

The traffic congestion is getting worse at urbanized cities around the globe



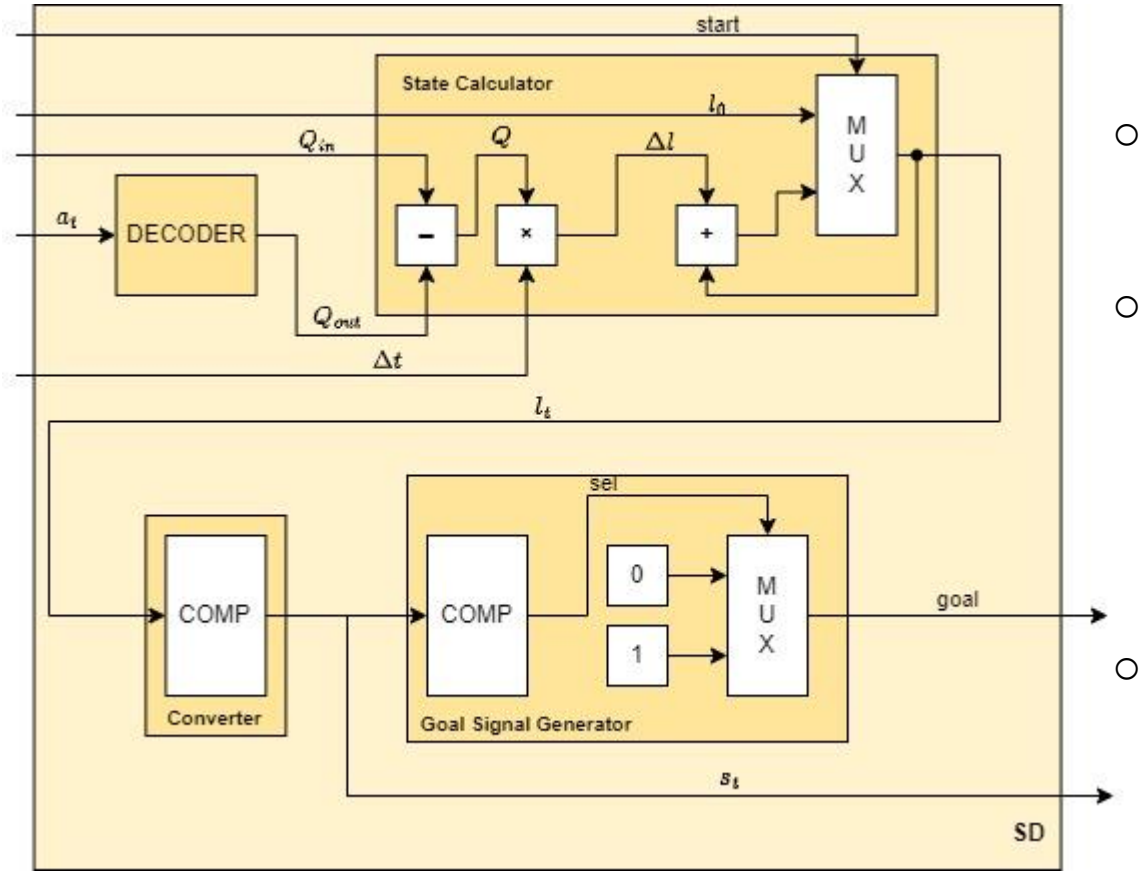
We need a hardware accelerator that enables machine learning to work faster

How do we model the Reinforcement Learning?



- We divide each road into **8 different levels**.
- It leads to $N_{state} = 8^4 = 4096$ state
- We design **4 different action** to control the traffic

How do we simulate the environment?



State decider architecture

- It simulates the environment for the agent.

$$l_t = l_{t-1} + (Q_{in} - Q_{out})(\Delta t)$$

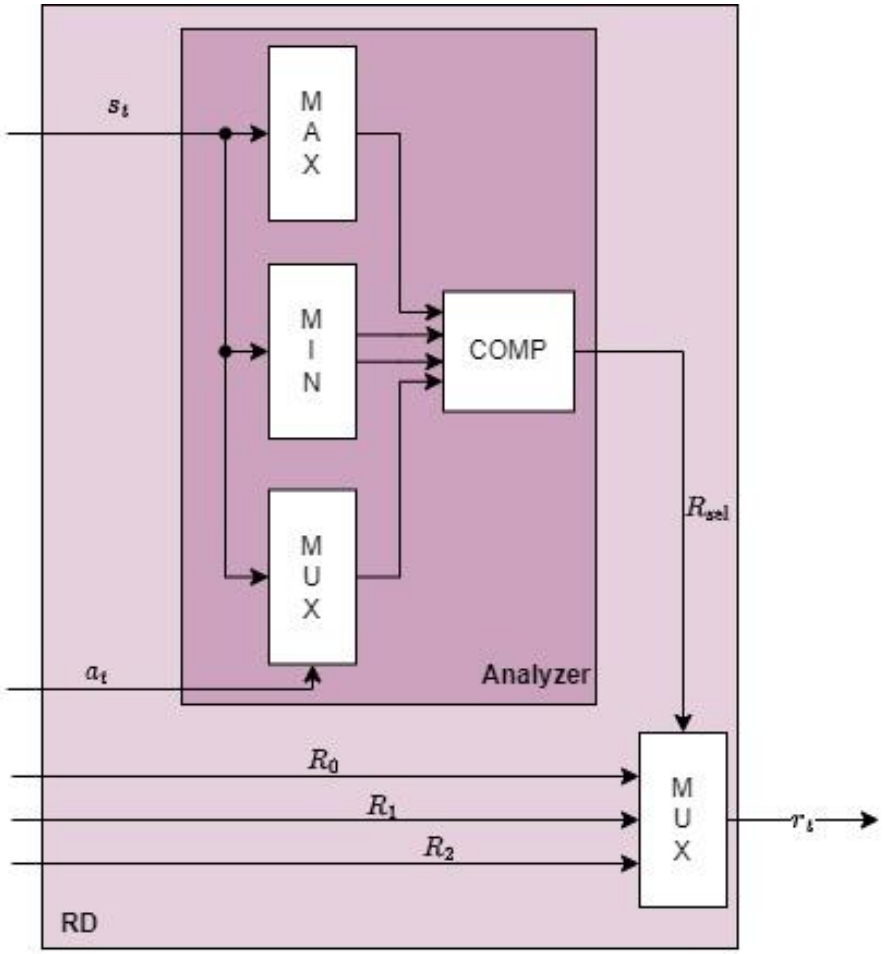
- Then we transform the traffic length into congestion level.

$$L_r = \begin{cases} 0, & if\ 0 \le l_t < b_0 \\ \vdots & \vdots \\ 7, & if\ b_6 \le l_t < b_7 \end{cases}$$

- The we convert the congestion levels into state

$$s_t = L_A + 2^3 L_B + 2^6 L_C + 2^9 L_D$$

How is the interaction between the agent and the environment?



Reward decider architecture

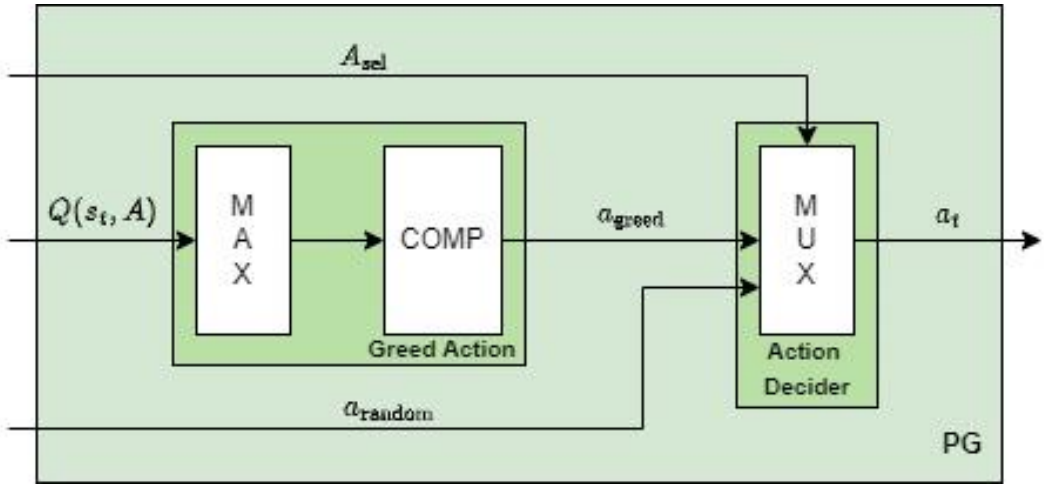
- Reward is the representation of how good the action taken by the agent in terms of controlling the traffic congestion.
- The reward is based on action.

$$R = \begin{cases} R_0, & \text{if bad action is taken} \\ R_2, & \text{if good action is taken} \\ R_1, & \text{otherwise} \end{cases}$$

Where the value for each R_x should fulfill the equation:

$$R_2 > R_1 > R_0$$

How do we control the policy?



Policy generator architecture

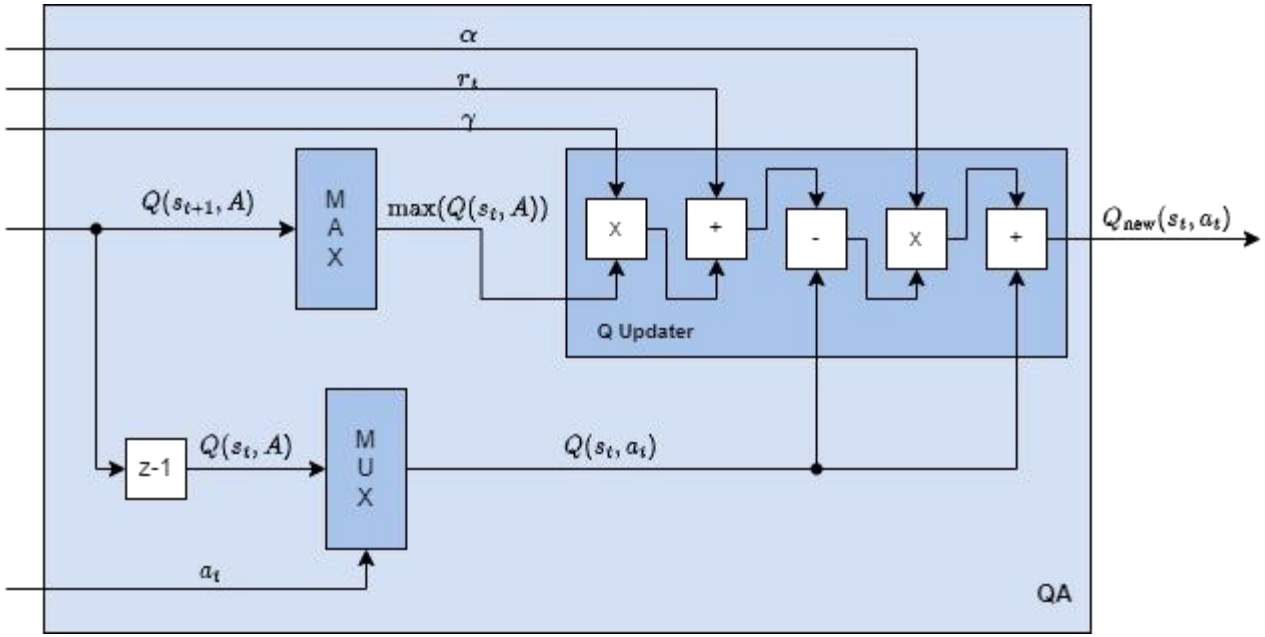
It controls the agent's behavior to take an action.

$$a_{t+1} = \begin{cases} \mathrm{a}_{\mathrm{greed}} & if\ x < \tau \\ \mathrm{a}_{\mathrm{random}} & otherwise \end{cases}$$

Where:

$$\tau = \Psi - \psi$$

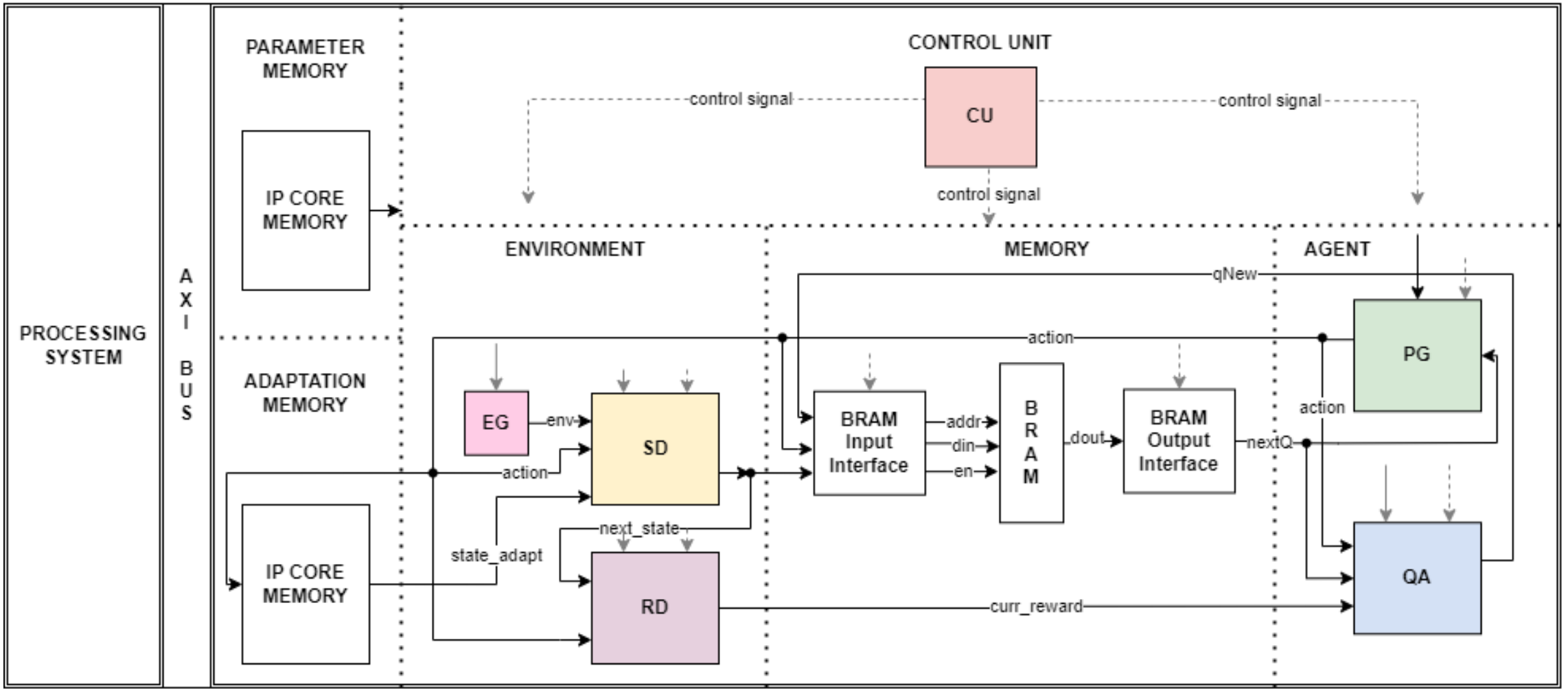
How does the agent learn?



It holds the bellman equation to update the Q-value.

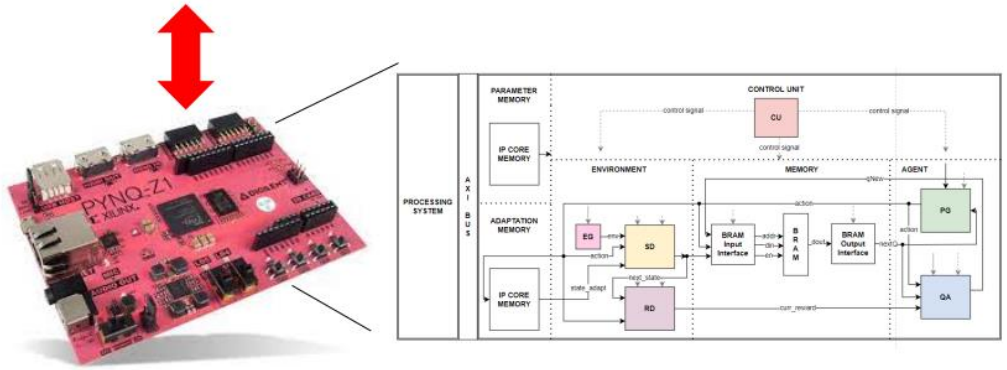
$$Q(s_t, a_t) = (1 - \alpha)\big(Q(s_t, a_t)\big) + \alpha(r_t + \gamma \times \max(Q(s_{t+1}, A)))$$

The hardware architecture!



How do we test the design?

Pygame Traffic Simulator
(Phyton SW)



Q-learning Accelerator on PYNQ-Z1 platform

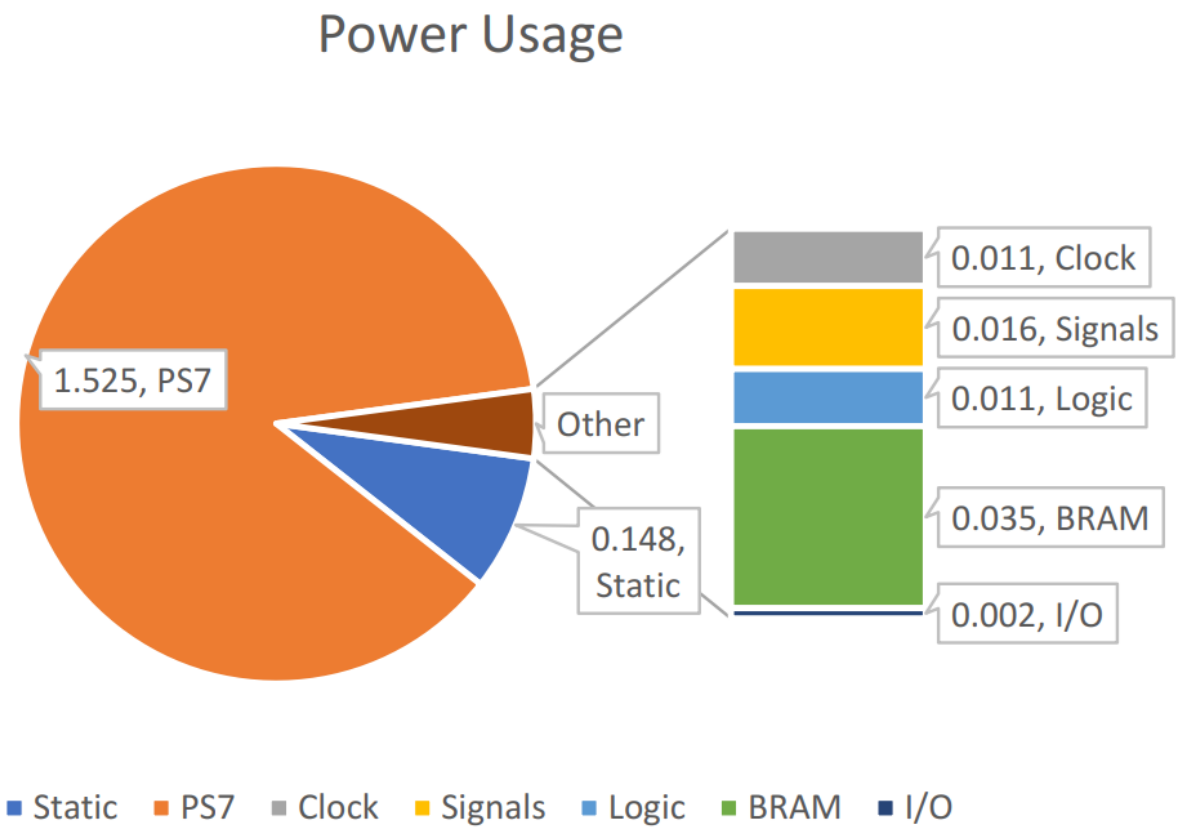
Parameter	Values
Environment size	4096×4
Learning coefficient α	0.825
Discount factor γ	0.125
Data representation	Fixed-point
Bit width [integer, fraction]	[16,16]

How is the chip performance?

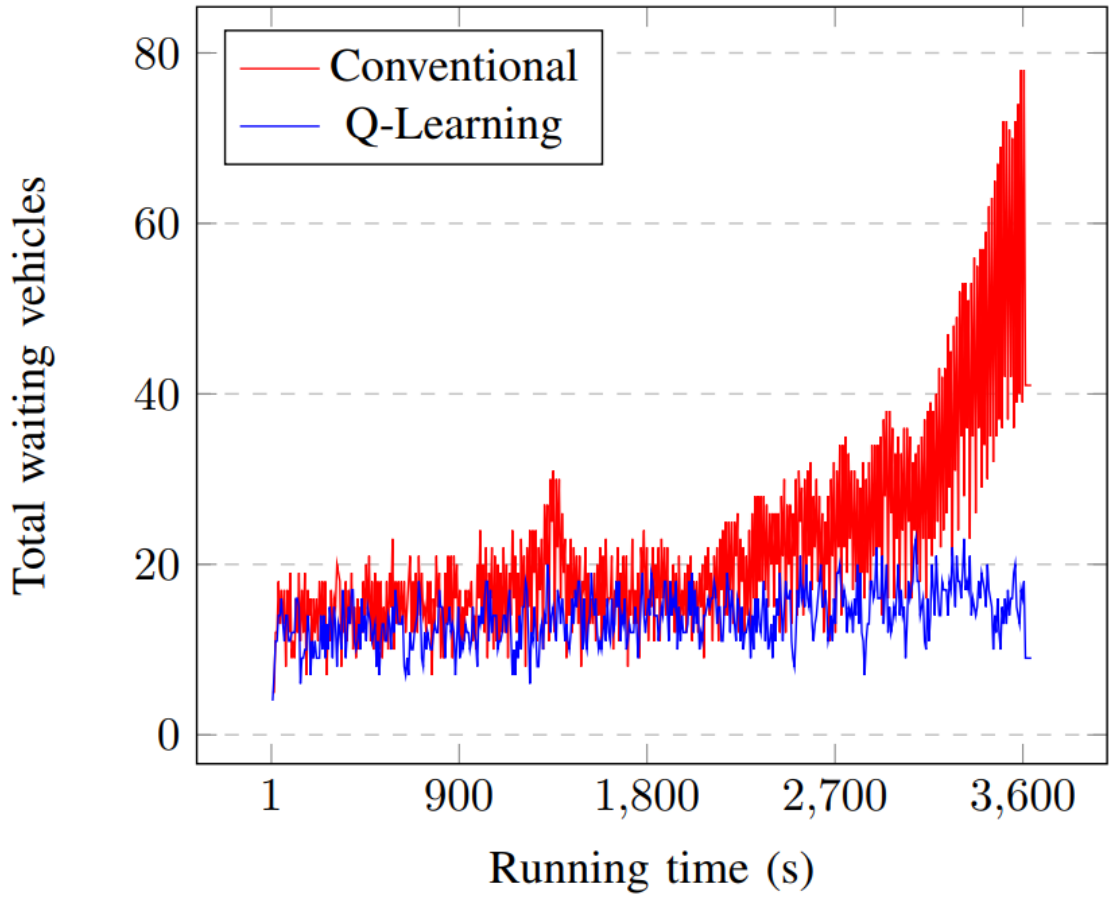
Timing variables	Quantity
Clock Frequency	100 MHz
Worst Negative Slack	1.153 ns
Worst Hold Slack	0.031 ns
Worst Pulse Width Slack	3.750 ns

Resource	Utilization		
	Used	Available	Percentage
Slice LUTs	5,200	53,200	9.77%
Slice Registers	6745	106,400	6.34%
F7 Muxes	125	26,600	0.47%
Slice	2,322	13,300	17.46%
LUT as Logic	331	17,400	1.90%
LUT as Memory	331	17,400	1.90%
Block RAM Tile	34	140	24.29%

Power usage

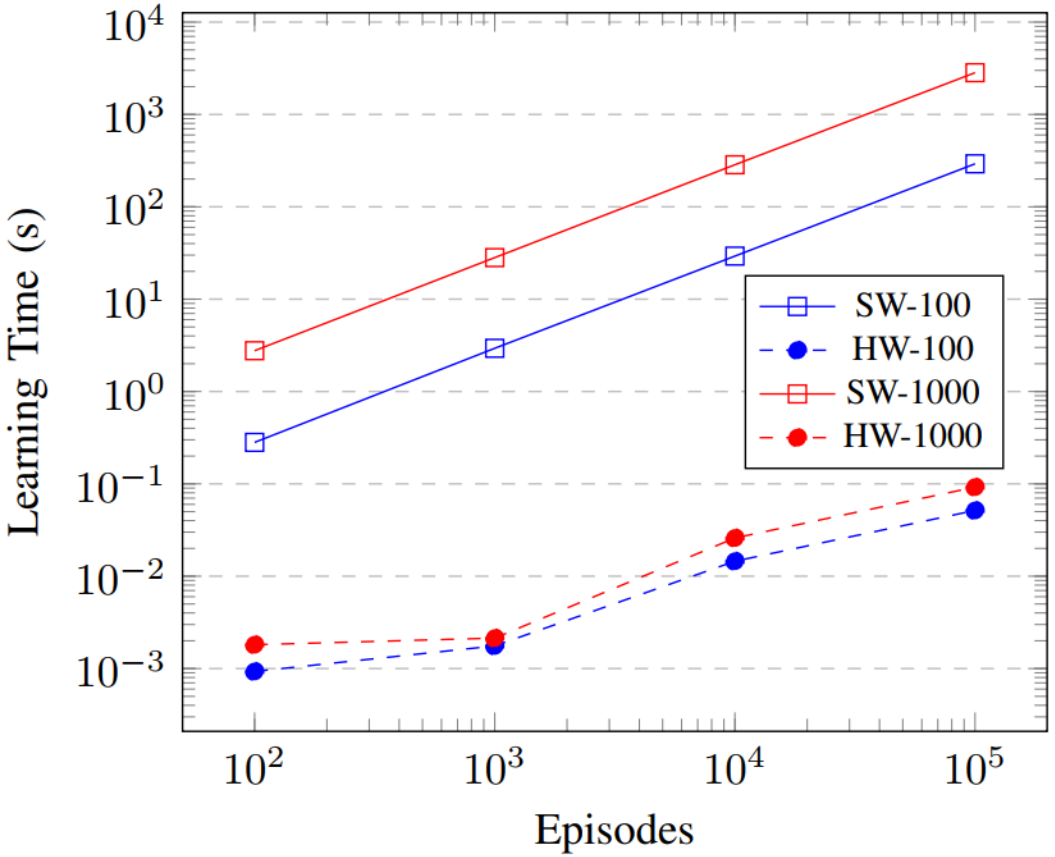

Power Usage
1.525, PS7
Other
0.148, Static
0.011, Clock
0.016, Signals
0.011, Logic
0.035, BRAM
0.002, I/O
Static
PS7
Clock
Signals
Logic
BRAM
I/O

How is the system performance?

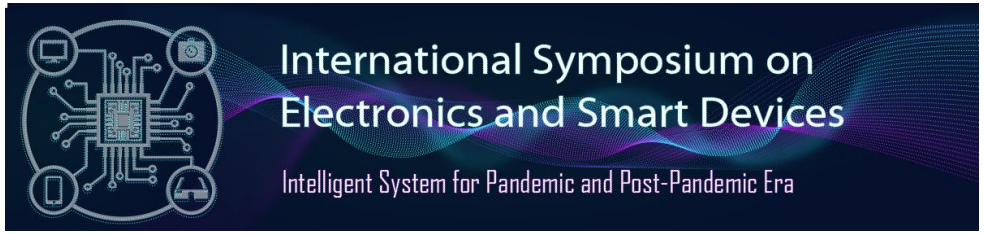


- We compare the performance of our methods w ith the fixed-time implementation
- We use python pygame to simulate both imple mentation.

How much does the system accelerate?



- We compare the system processing time with the software implementation.
- The software runs on PC with AMD Ryzen 5 3600 with 8 GB RAM



What question do you have?

2022 International Symposium on Electronics and Smart Devices
November 8-9, 2022, Hybrid Conference, Bandung