



國立陽明交通大學

National Yang Ming Chiao Tung University

**AI + SDR = Intelligence Radio :**

**運用人工智慧模型於 RFSoC 實現之智慧感知無線電**

Presenter: 林孟熹、歐軒佑

Advisor: 王蒞君 教授

National Yang Ming Chiao Tung University, Taiwan

# BACKGROUND

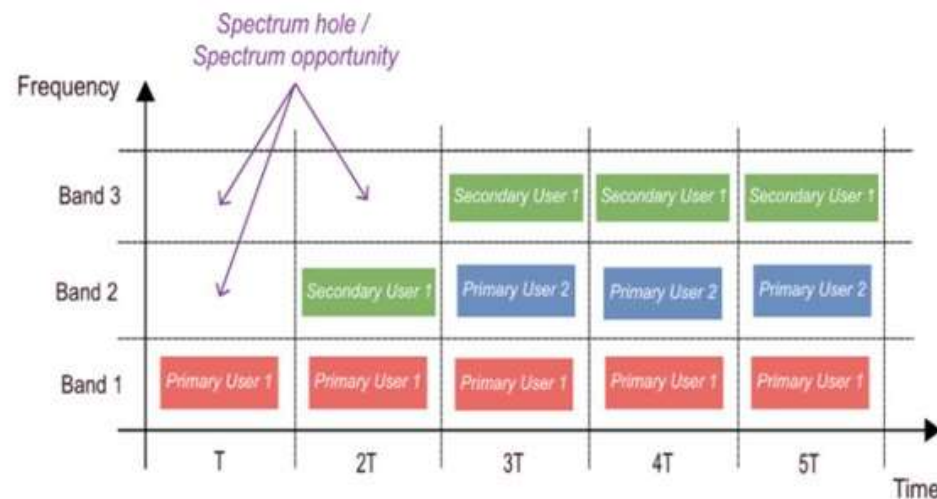
- 頻譜感測(Spectrum Sensing)：

- 尋找空閒的頻段，能讓該頻段給用戶使用  
增加頻譜使用效率

- 研究動機：

- 隨著IOT設備的增加導致頻譜資源可能面臨嚴重短缺。為了解決這個問題，頻譜檢測技術極為重要

- ✓ 高效率的頻譜感測：**快速運算、高準確率**



# OUTLINE

1. SDR on RFSoc
2. 感知無線電(Cognitive Radio)& 動態頻譜存取(Dynamic Spectrum Access)
3. DSA: Spectrum Sensing using Green Learning & Deep Learning
4. DSA: Frequency Hopping
5. 智慧感知無線電(Intelligence Radio)

1

## SDR on RFSoc

# SDR on RFSoc

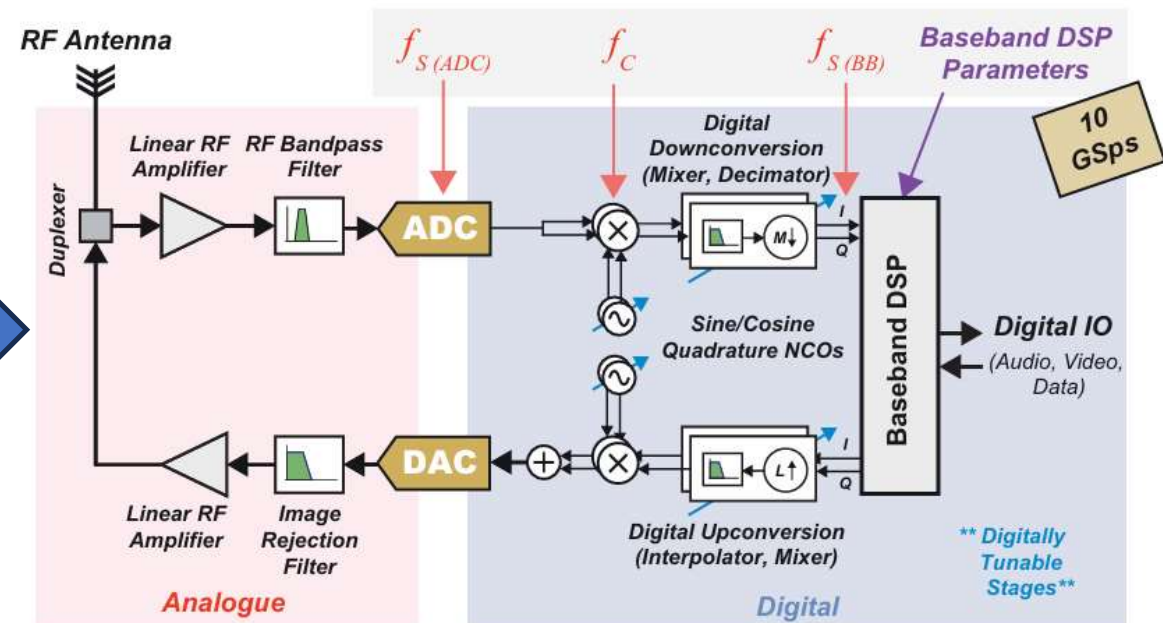
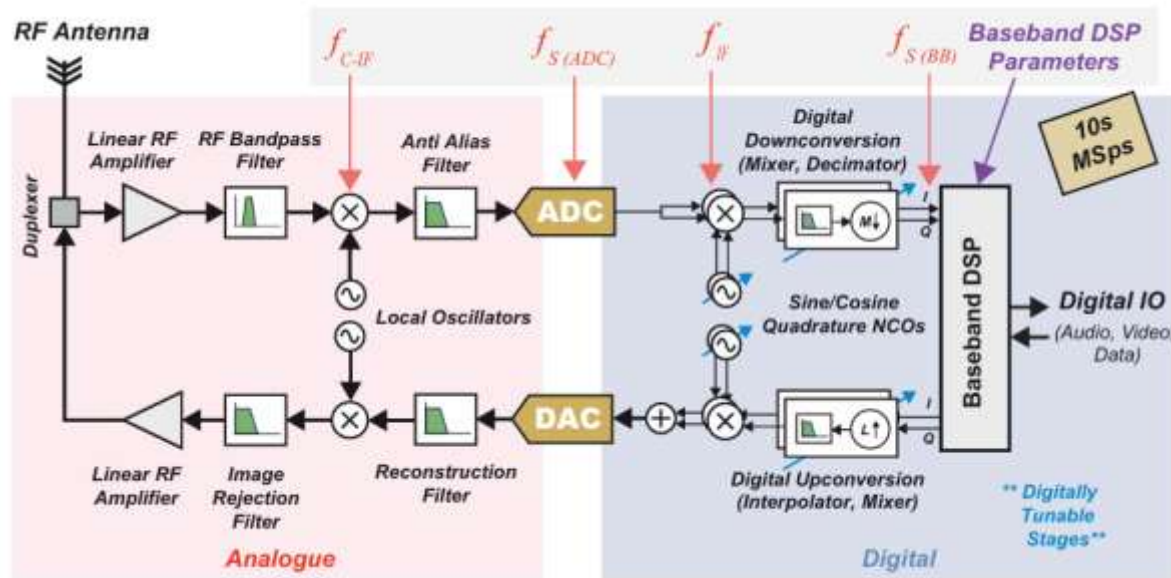


軟體無線電(Software Defined Radio, SDR)為 *Joseph Mitola* 於1990年代提出之開創性新概念，將傳統通訊設備中之特殊應用積體電路(ASIC)以**可程式化邏輯電路(FPGA)**替代，使其**通過軟體即可修改架構**，大大增加應用彈性及減少更新換代時因硬體無法修改造成的浪費。

其中，由AMD推出的Zynq RFSoc系列在性能及易用性上都十分卓越，是實現SDR的理想平台。

# SDR on RFSoc

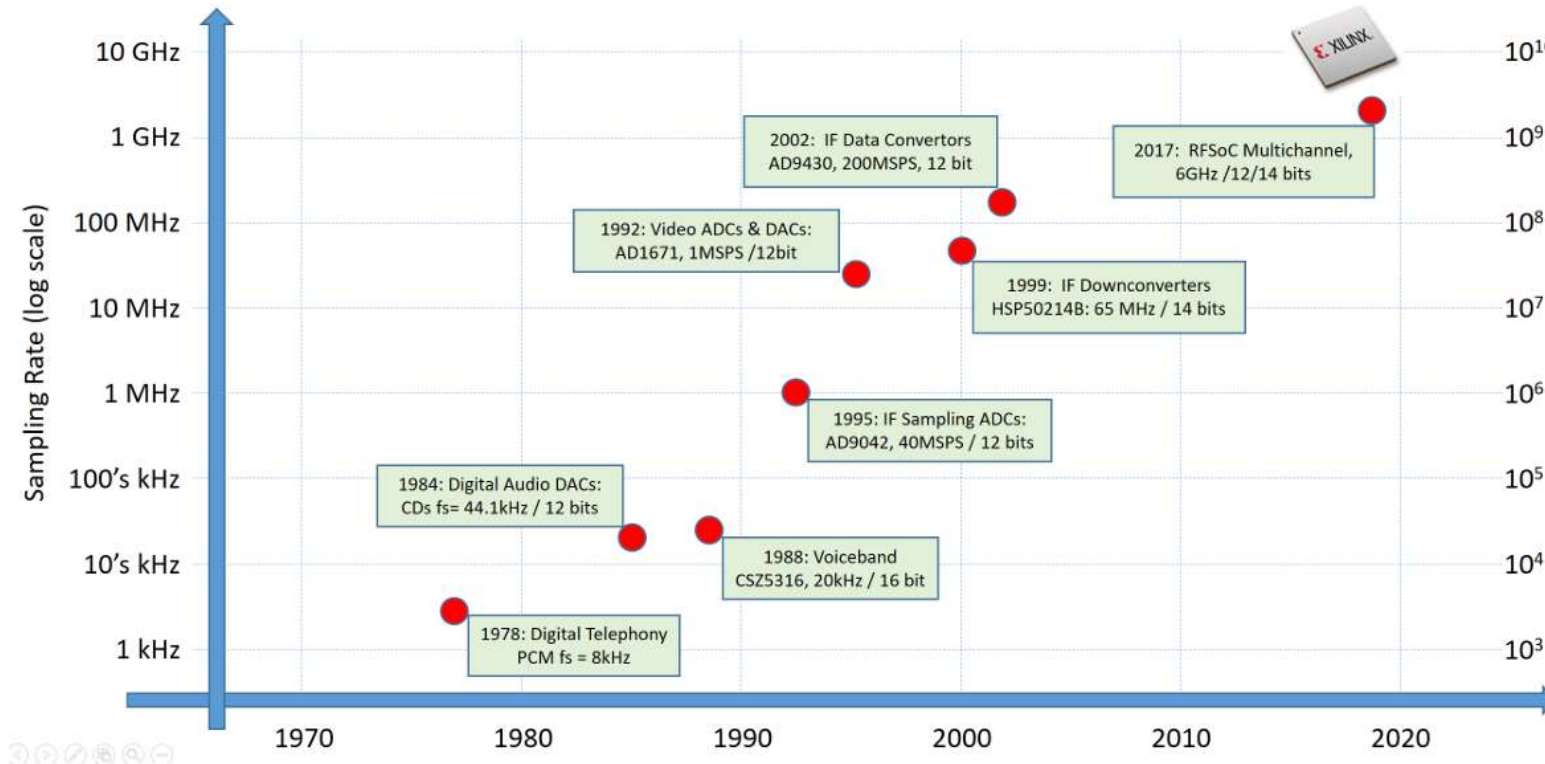
- SDR平台實現的關鍵，在於類比數位轉換器(ADC&DAC)採樣率的提升，因其代表了能以數位方式處理的頻率範圍。當採樣率高到不需要類比混頻電路時，才能夠動態調整訊號發送之頻率，也就是動態頻譜存取(Dynamic Spectrum Access)。





# SDR on RFSoc

- 當採樣率達到10GHz級別時，裝置可以直接採樣並處理大部分通訊頻率之訊號，因此具備7.125GHz採樣率的 RFSoc 已能實質作為現今通訊標準下之SDR平台。



- Radio: 3 kHz to 3 GHz
- Microwave: 3 GHz to 300 GHz

## Zynq RFSoc DFE



**7.125GHz**  
of Analog Bandwidth

# SDR on RFSoc

相較於其他SDR，RFSoc還具有以下優點/應用:

- 具有基於Python的開發環境PYNQ，簡化FPGA開發流程，降低開發門檻。
- 可在單一裝置上同時部屬多種無線通信協定，如WiFi、藍芽、4G/5G。
- 可作為新通訊協定/裝置開發、測試之平台。
- 因其功能之強大、體積之小，結合無人飛行載具(UAV)使用可作為高機動性之行動基地台。
- 使用者可在FPGA自由部署各種需要高速硬體電路支援之算法，如AI模型、DSP模塊。

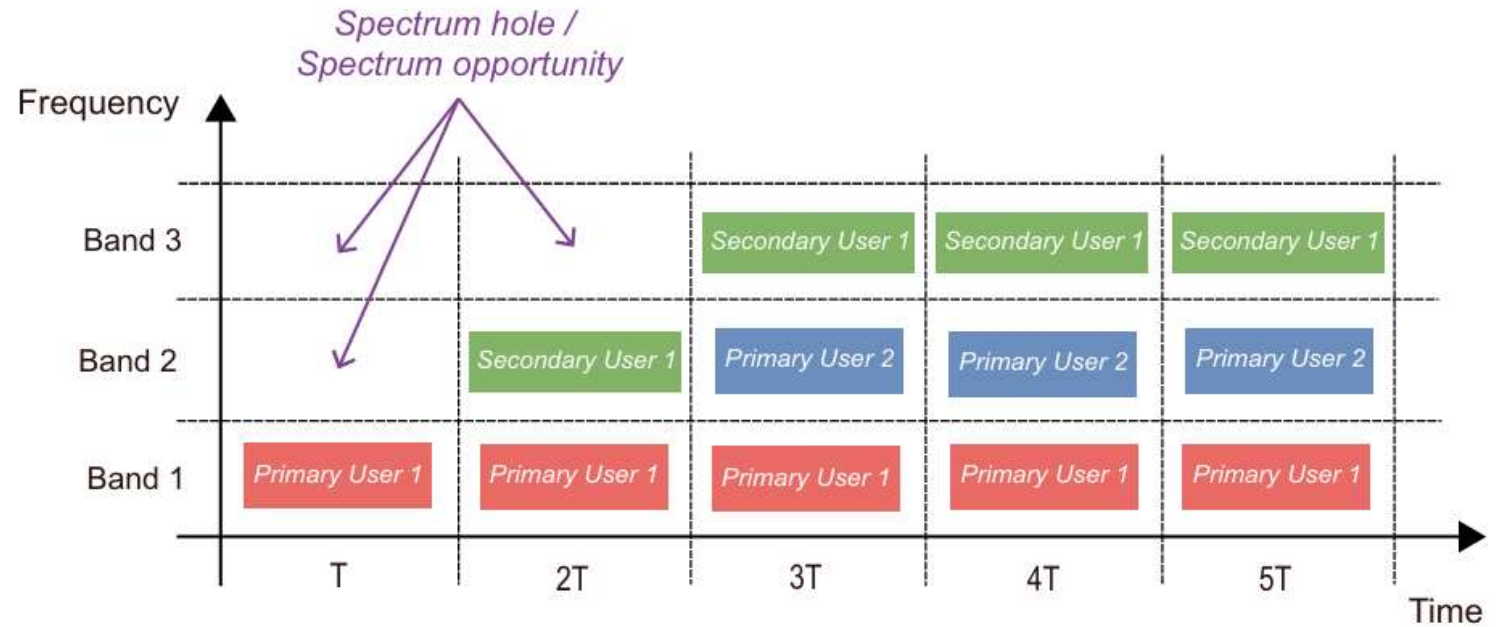
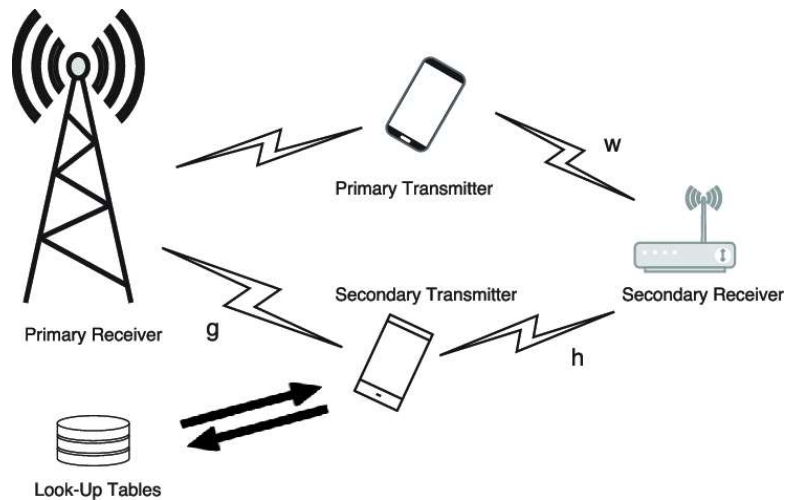


2

## 感知無線電 & 動態頻譜存取

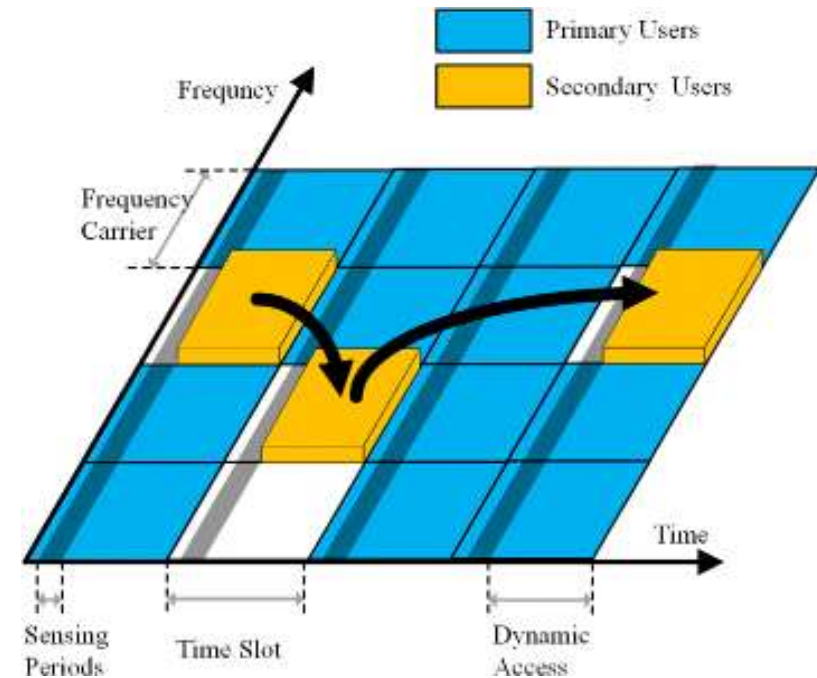
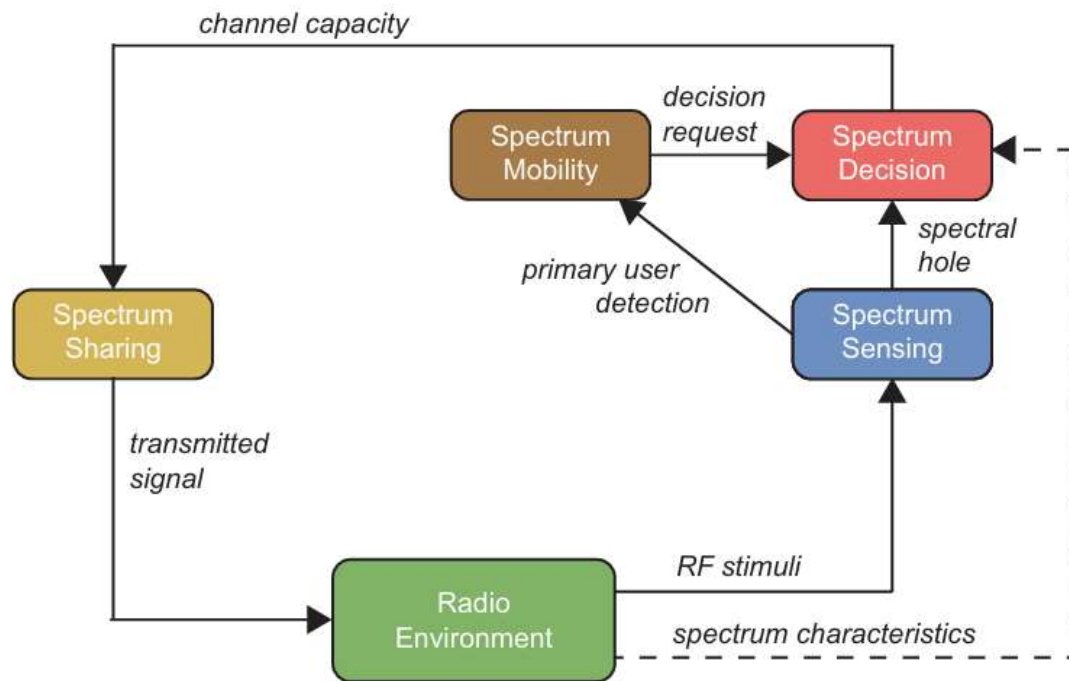
# 感知無線電(Cognitive Radio)

感知無線電(Cognitive Radio)由 Joseph Mitola(SDR提出者)提出，是利用閒置頻譜，即頻譜空洞 ( Spectrum Holes ) 進行動態頻譜存取 ( Dynamic Spectrum Access ) 與傳輸，以達到在擁擠頻譜空間進行訊息傳輸，提高頻譜利用率的技術。



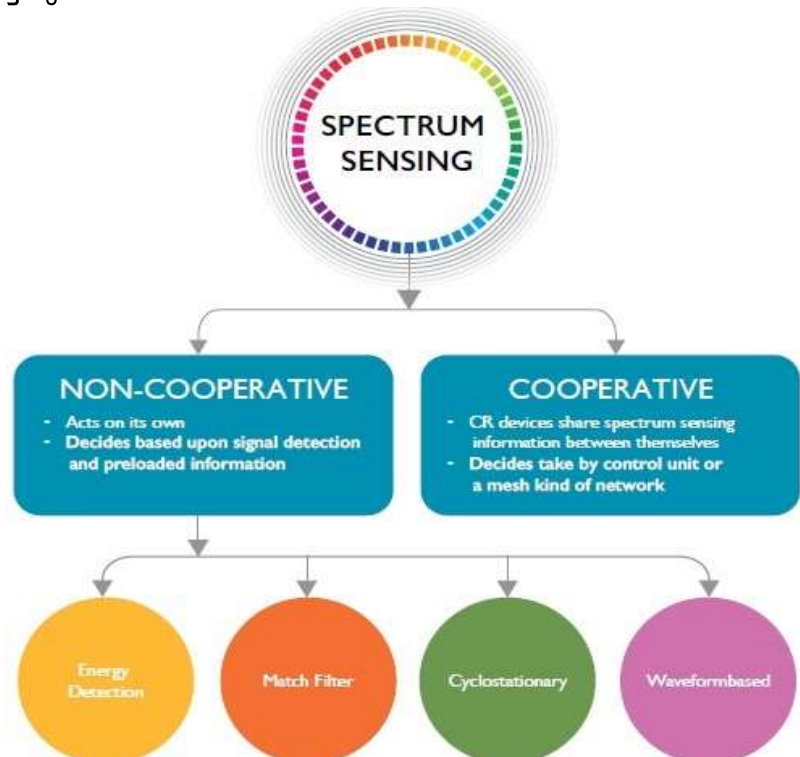
# 動態頻譜存取 ( Dynamic Spectrum Access )

要進行動態頻譜存取，裝置至少須具備兩個功能: 頻譜感測(Spectrum Sensing)及跳頻(Frequency Hopping)。頻譜感測辨識頻譜空洞，跳頻改變裝置傳輸頻率至該空洞。



# 如何辨識頻譜空洞？

頻譜感測(Spectrum Sensing)有許多固定算法，如能量檢測、循環平穩特徵檢測、匹配濾波等等，但這些算法皆有其適用與不適用之場景，因此較為理想的做法是使用**機器學習**方法訓練模型。



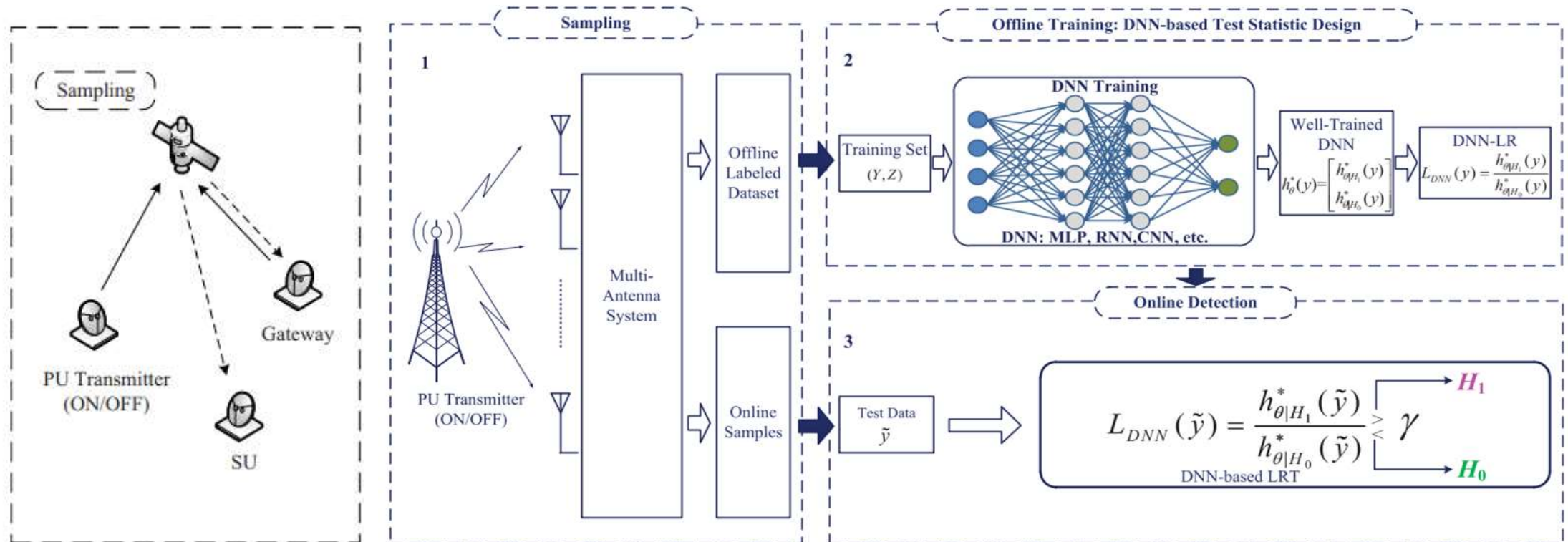
方法	準確度	噪聲敏感性	適用場景
能量檢測	中	高	噪聲穩定、硬件簡單的場景
循環平穩特徵檢測	高	低	信號 <b>特徵已知</b> 的應用
匹配濾波	非常高	低	<b>信號已知</b> 的高要求場景
機器學習	高	低	<b>數據豐富</b> 的場景

3

## Spectrum Sensing using Machine Learning

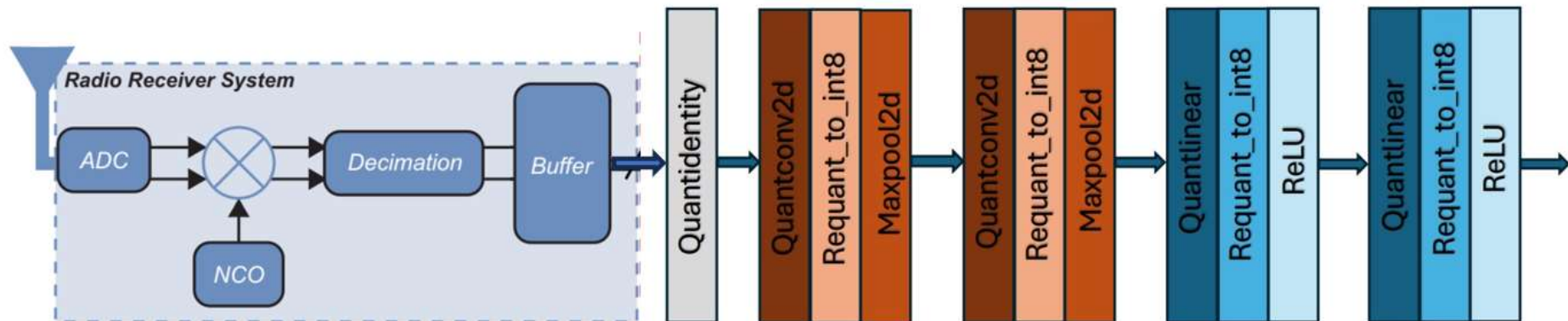
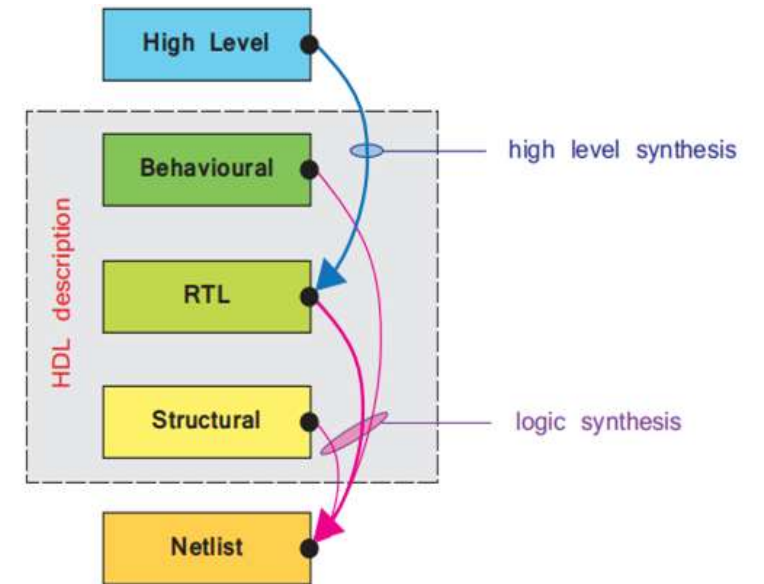
# 訓練流程

訓練頻譜感測模型需要先收集大量訊號資料做offline Training，再將訓練好的模型放到RFSoc的FPGA上，讓它能夠對收到的訊號實時進行分類 ( $H_0$ 表示閒置， $H_1$ 表示已佔用)。



# 系統架構與流程

- Step 1: Generate training dataset (Python)
- Step 2: Model building (Python)
- Step 3: Translate Python  $\rightarrow$  C++  $\rightarrow$  HDL
- Step 4: Digital Circuit Design for the Model (Vivado)
- Step 5: Radio Transceiver System (RFSoc)
- Step 6: Prediction Results

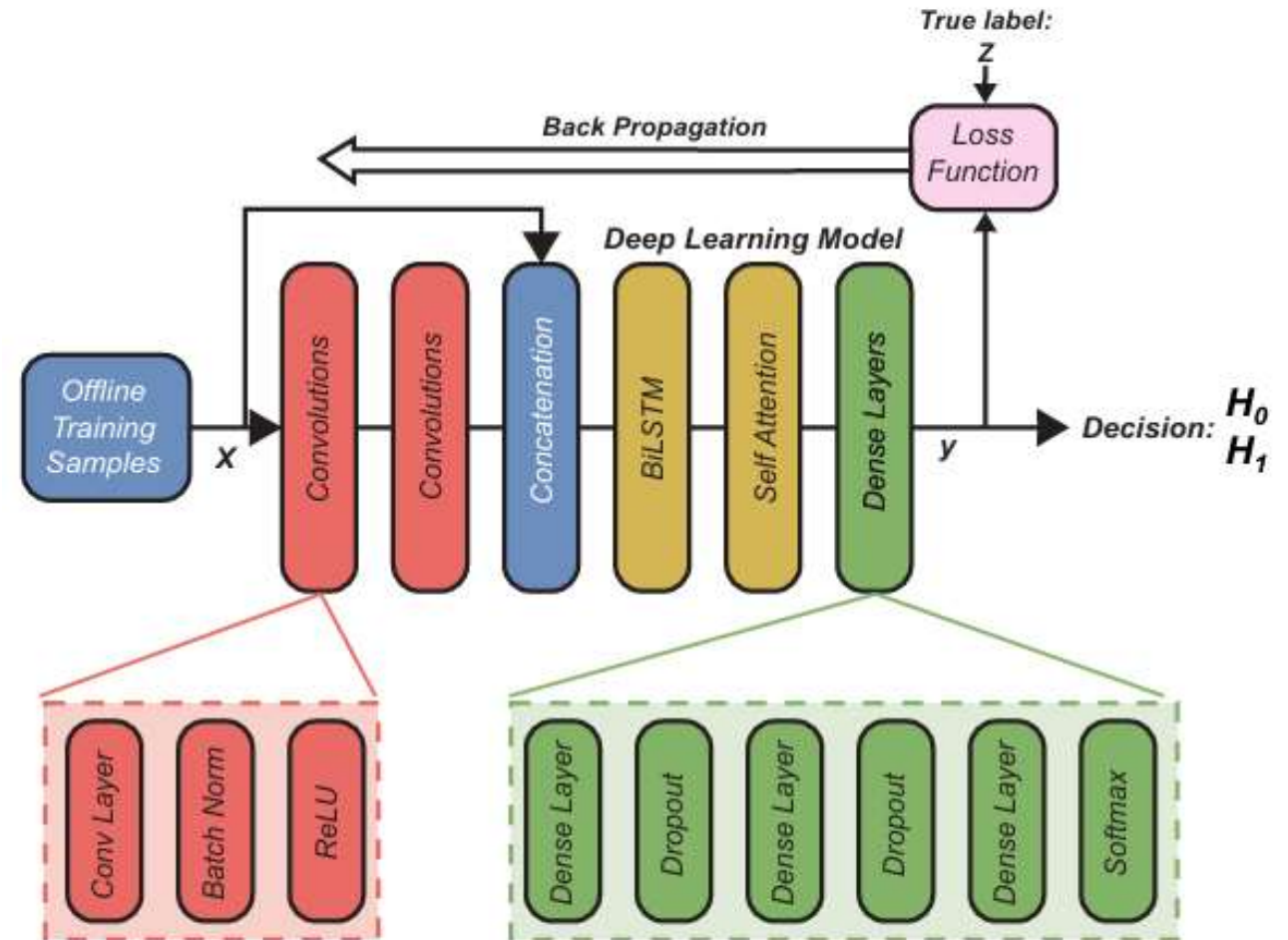




# Deep Learning Model

一個理想的頻譜辨識深度學習模型結構如右圖，包含：

- 1.卷積層(CNN): 提取訊號之**瞬時特徵**。
- 2.雙向長短期記憶 (BiLSTM): 識別卷積層提取之瞬時特徵的**前後雙向時間關係**。
- 3.全連接層(Dense Layers): 將學到的特徵轉化為**分類結果**。



# Challenges faced by RNN and CNN

- **訓練資料需求**

獲取頻譜感知任務的資料可能很困難。

- **計算複雜度**

CNN和RNN具有較高的計算負擔，尤其是在計算資源有限的設備上。

- **延遲問題**

可能無法做到Real-time Reaction。

- **硬體需求**

對於存儲空間有限的小型設備來說，這可能會造成負擔。

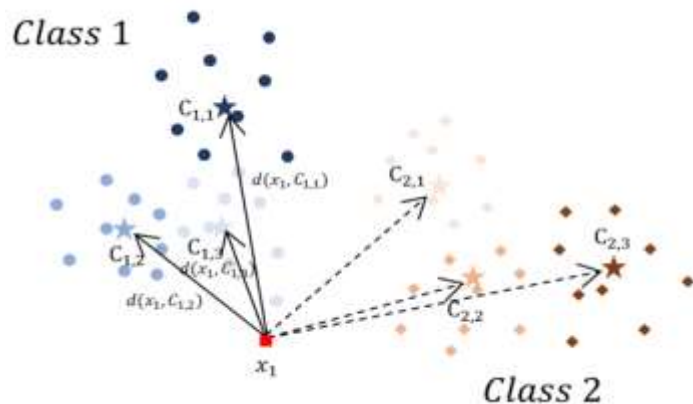


**Green Learning Model can fix!!!**

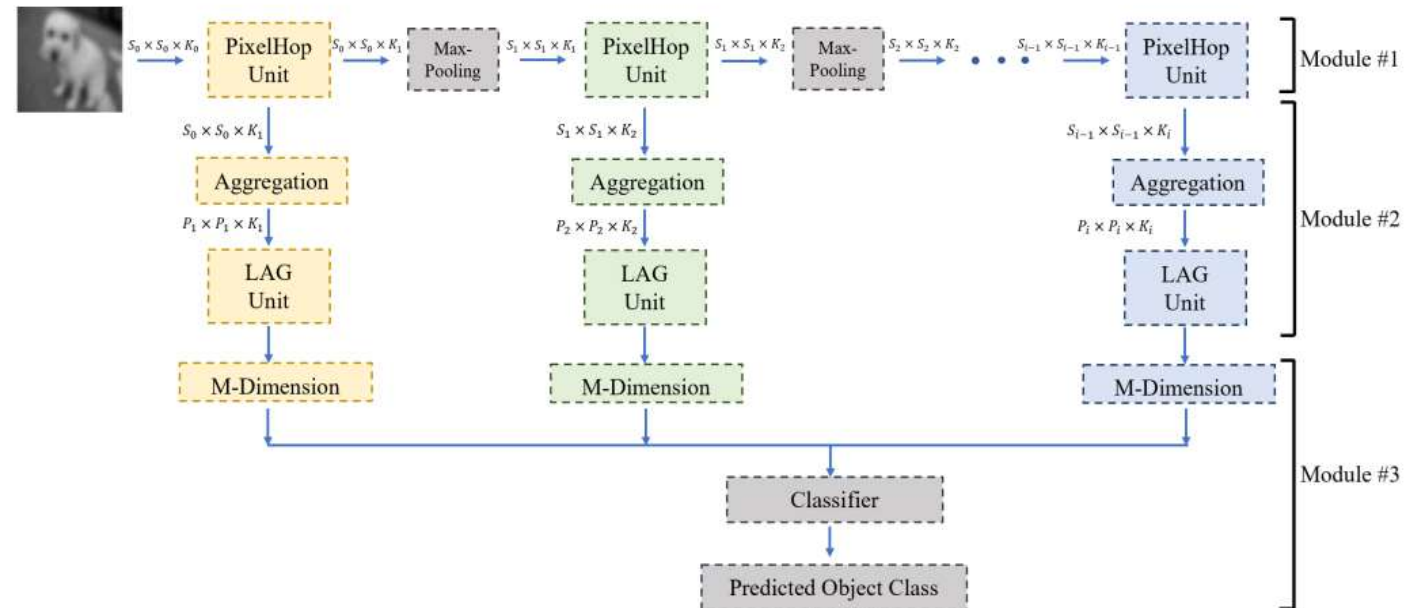
# Green Learning: PixelHop Model

Pixelhop 是不使用任何的神經元的結構。

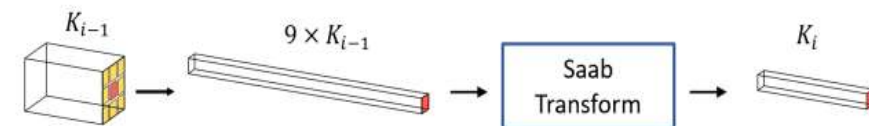
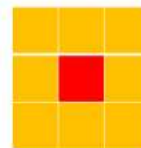
它主要通過對數據進行多次 PCA 來簡化特徵，將特徵映射到低維度上，再使用 SVM 進行分類。



▲ Lag unit



▲ Pixelhop model



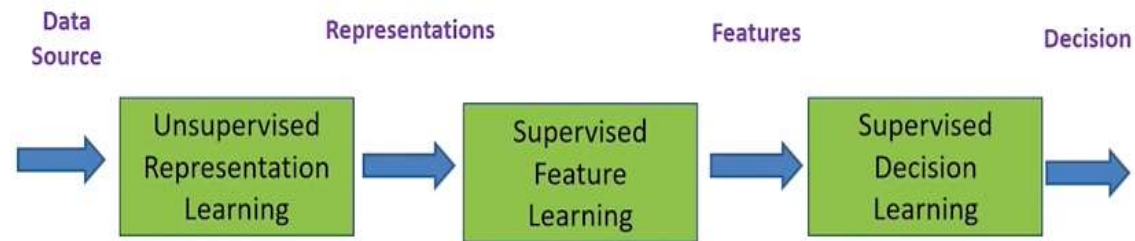
▲ Pixelhop unit

# Green Learning 的優勢



## 與DL的差別

- 不使用神經元
- 不使用神經網路做傳播
- Model 的 size 較小 (大概為 DL model 的 1%)
- 計算複雜度低 (大概為 DL model 的 0.1%)
  - ✓ 低碳足跡 (Energy-saving)
- 計算速度快



▲ Overview of Green Learning

# 訓練資料

Generate signals under different SNR levels :

- QPSK Signal (Signal + Noise): Label 1
- Noise: Label 0

## Training Features

- Energy Feature
- Power Spectrum Feature
- Cyclostationary feature

- Energy Feature:

$$T(y) = \frac{1}{N} \sum_{n=1}^n |y(n)|^2$$

- Power Spectrum Feature:

$$S_x(k) = \frac{1}{N} |X(k)|^2 = \frac{1}{N} |FFT[x(n)]|^2, \quad k = 0, 1, \dots, N-1$$

- Cyclostationary Feature:

$$\begin{aligned} R_x(t, \tau) &= E\{x(t)x(t-\tau)^*\} & R_x(t, \tau) &= R_x(t+T_0, \tau) \\ R_x(\alpha, \tau) &= F(R_x(t, \tau)) & S_x(\alpha, f) &= F(R_x(\alpha, \tau)) \end{aligned}$$

# Green Learning VS DL(CNN)

## ▼ Green Learning in Computer Simulation

Number of Data:" 2080

Prediction Time: 0.2110 seconds

Prediction Time pre data: 0.00010 seconds

Layer (type)	Output Shape	Param #
PixelHop	[9, 2x2]	36
LAG (Clustering)	[20, 36]	720
SVM	[36, 1]	37
Total params: 793		
Trainable params: 793		
Non-trainable params: 0		

## ▼ Deep Learning in Computer Simulation

Number of Data: 65

Prediction Time: 0.0790 seconds

Prediction Time per data: 0.0012 seconds

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 4, 65, 5]	20
MaxPool2d-2	[-1, 4, 64, 4]	0
Conv2d-3	[-1, 8, 33, 3]	136
MaxPool2d-4	[-1, 8, 32, 2]	0
Linear-5	[-1, 128]	65,664
Sigmoid-6	[-1, 128]	0
Linear-7	[-1, 84]	10,836
Sigmoid-8	[-1, 84]	0
Linear-9	[-1, 48]	4,080
Sigmoid-10	[-1, 48]	0
Linear-11	[-1, 2]	98

Total params: 80,834

Trainable params: 80,834

Non-trainable params: 0

Compared to DL, GL has advantage on....

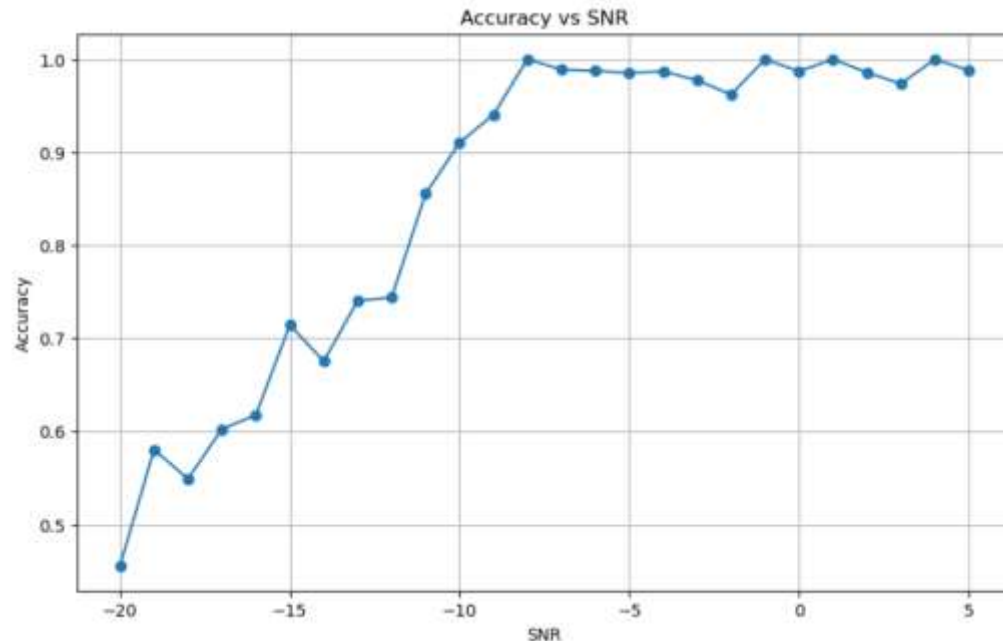
- Lower Param : Energy-efficient, low in computational complexity
- Faster prediction time : low latency, Real-time

# Green Learning VS DL(CNN) in Simulation

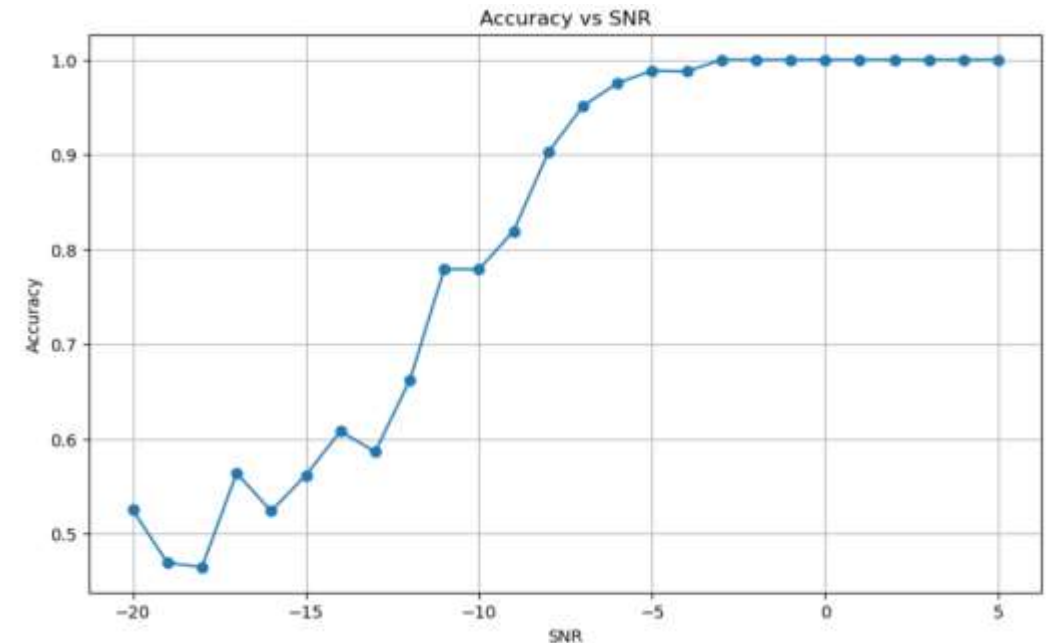
With PixelHop Green Learning method, the system is more **energy-efficient** and lower the **computational burden** and **latency**!

Green Learning demonstrates **higher accuracy than DL at low SNR levels**, but its accuracy becomes more variable as the SNR increases.

▼ Green Learning in Computer Simulation



▼ Deep Learning in Computer Simulation





# Green Learning VS DL(CNN) in RFSoc

After completing transforming Python into C++ and using vivado to transform into Verilog, we can run the model on FPGA board. The runtime for DL model decrease from 0.0012 s/data to **0.0002 s/data**.

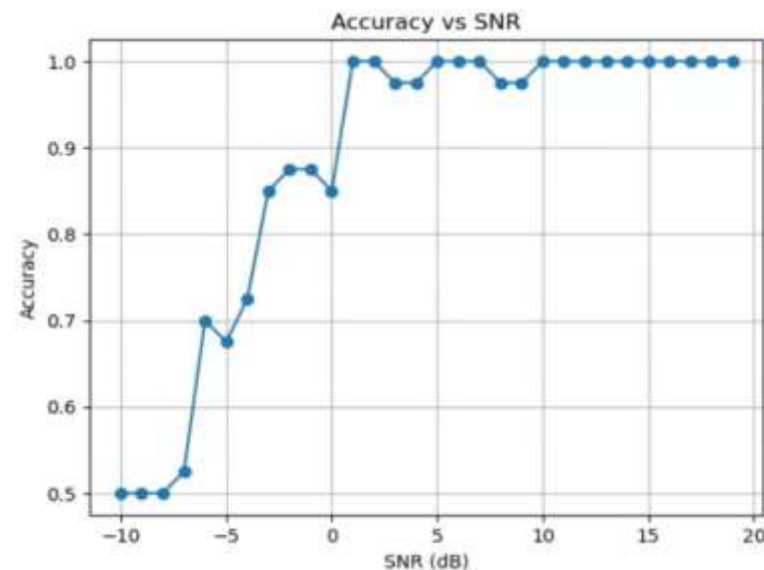
We believe that Green Learning can achieve a better performance than DL, but it still need to be verified.

But we haven't finish Green Learning in RFSoc Simulation,

```
print(f"準確率: {accuracy * 100:.2f}%")
```

單筆DATA的執行時間: 0.0002078413963317871  
準確率: 96.67%

▲ Deep Learning runtime in RFSoc Simulation



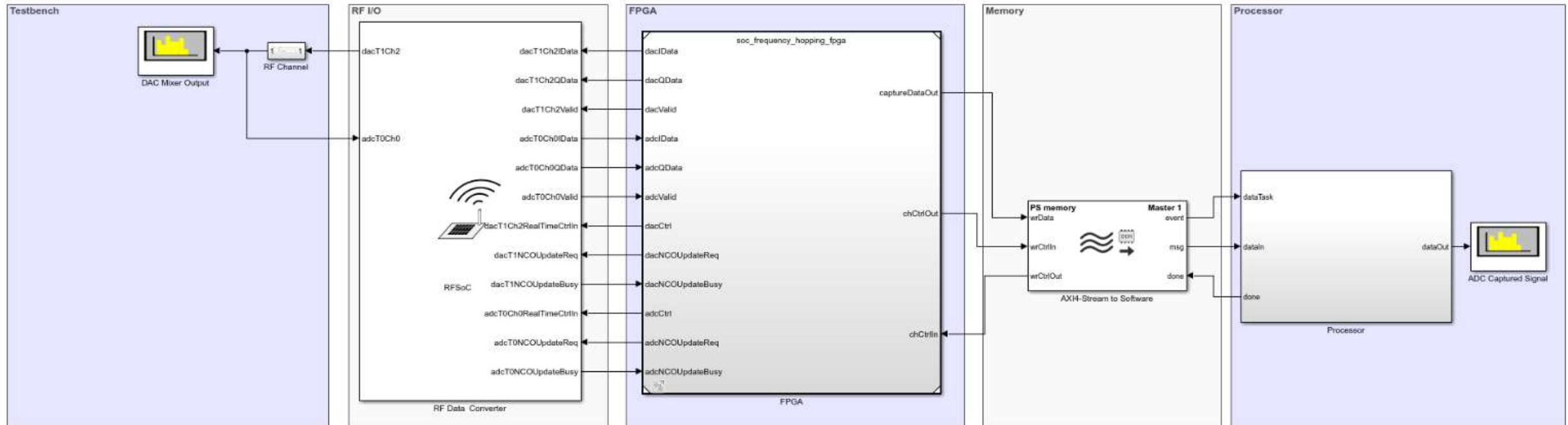
▲ Deep Learning in RFSoc Simulation

4

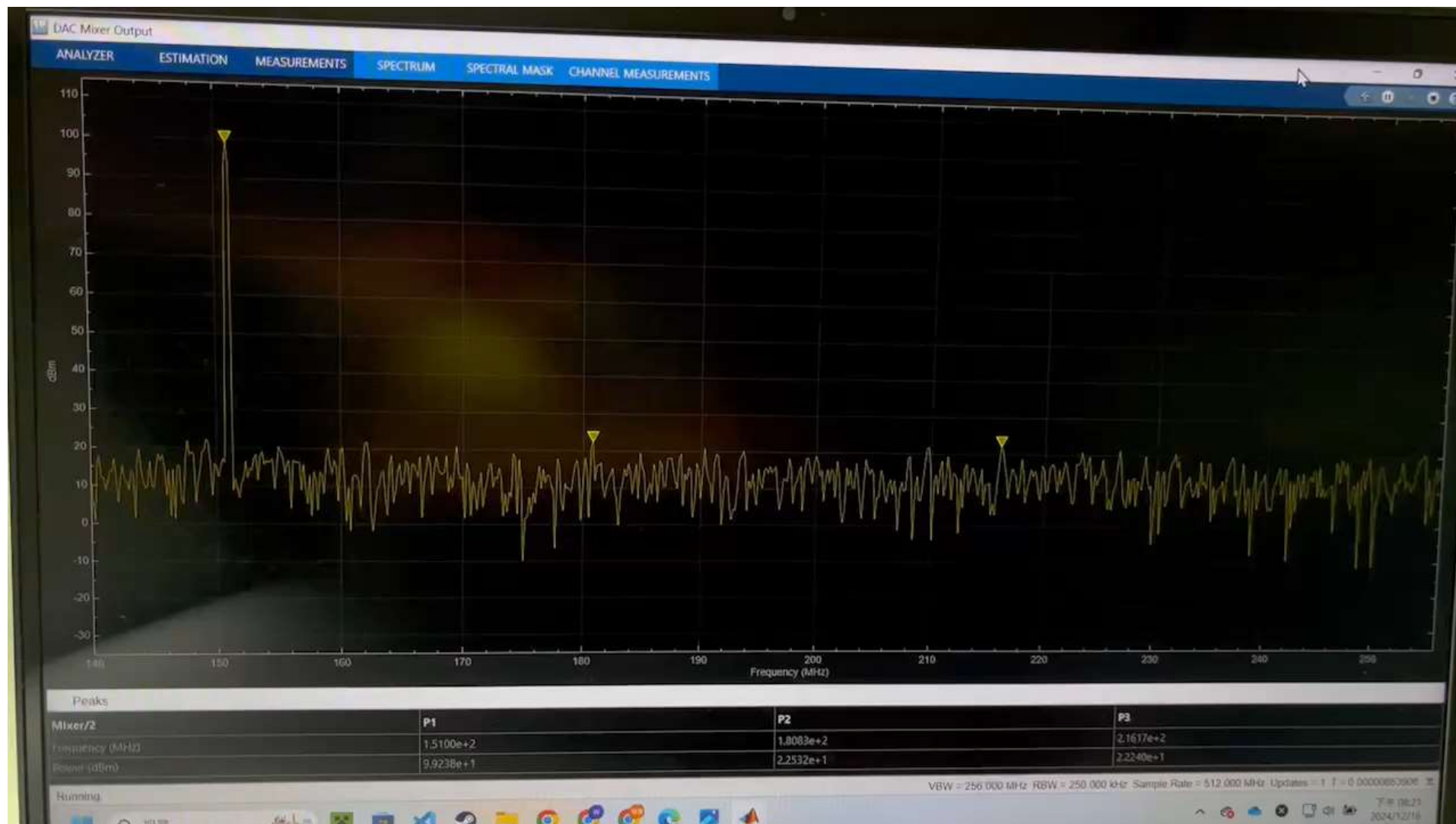
## Frequency Hopping

# Frequency Hopping

## Frequency Hopping on RFSoc Device



# Frequency Hopping

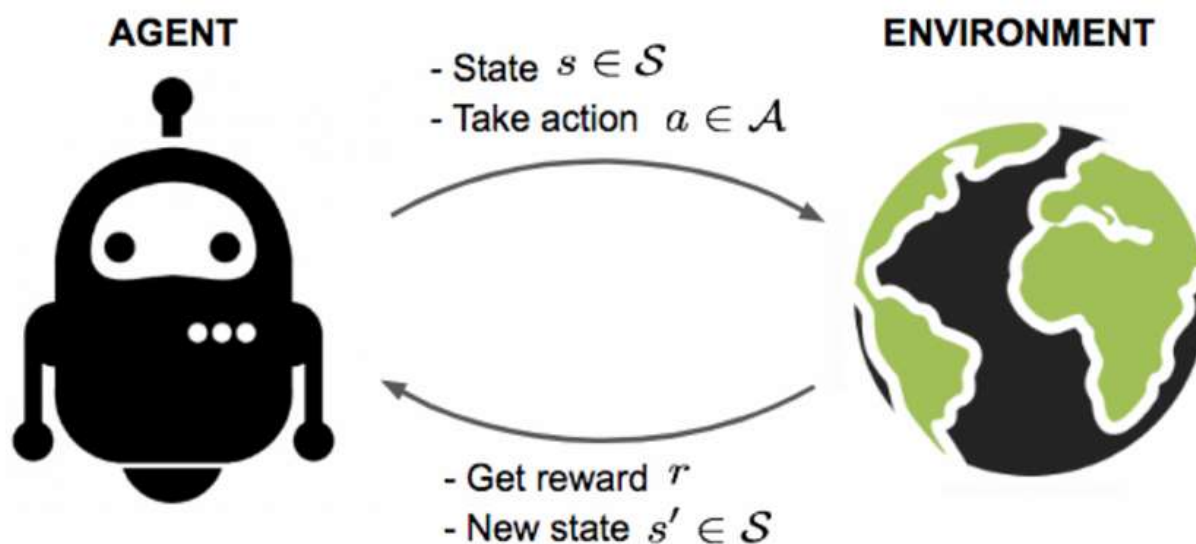


5

## 智慧感知無線電(Intelligence Radio)

# Future outlook

結合上述的頻譜感測及跳頻功能，目前RFSoc已能初步實現Joseph Mitola所提出的感知無線電，然而，要使通訊裝置真正具備全自動通訊能力，還需要再加入許多功能，例如: 多頻段感測、信道估計、自適應調制/編碼、抗干擾、多用戶協調。而除了針對每個功能訓練機器學習模型，還需要一個能夠統一考量所有因素的代理人，也就是需要運用**強化學習**的技巧。



# 智慧感知無線電

除了前面提到的動態頻譜存取及強化學習外，未來還會出現各種應用AI的場景，而要能夠發揮AI的強大潛力，就必須使用支援SDR的平台，因此SDR對於實現**智慧感知無線電**及**智慧物聯網**來說必不可少，這些都是RFSoc能夠發展的方向。

