

2025 Kentech 에너지 공학 탐구 동아리
동아리 FINDER(KE)

디지털 회로를 통한 MLP 구현

최성민 ○ 황준성

Sung-min Choi, Jun-seong Whang

대전대신고등학교

Daejeon Daeshin High School

요약

본 보고서는 퍼셉트론과 다층 퍼셉트론(MLP)의 개념을 디지털 회로로 구현함으로써, 인공지능 알고리즘이 하드웨어에서 어떻게 동작하는지를 근본적으로 학습하고자 한다. 연구팀은 기본적인 산술 연산 회로(가산기, 감산기, 곱셈기, 나눗셈기)를 바탕으로 퍼셉트론 구조를 회로화하고, XOR 문제와 Iris 데이터셋 이진 분류 문제를 MLP 구조로 설계해 구현하였다. 특히, 소프트웨어에서 학습한 가중치와 편향을 고정소수점으로 변환해 회로에 반영함으로써 AI 모델의 하드웨어 이식 가능성을 탐색하였다. 이를 통해 로지심(Logisim)을 활용한 시뮬레이션과 실제 동작 과정을 통해 디지털 회로 수준에서의 인공지능 구현 원리를 구체적으로 이해할 수 있었다.

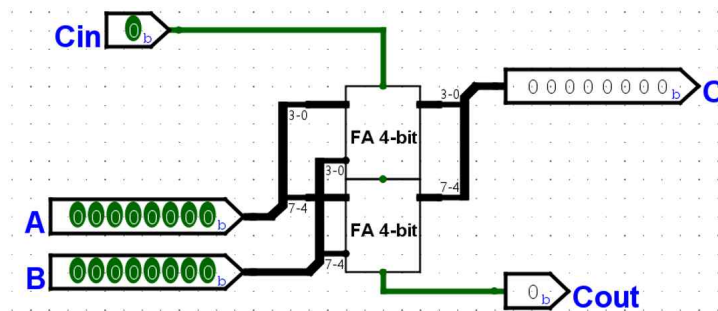
I 서론

가) 연구 목적

본 연구의 목적은 퍼셉트론과 다층 퍼셉트론(Multi-Layer Perceptron, MLP)의 원리를 바탕으로 인공신경망 구조를 디지털 회로로 설계하고, 이를 로지심(Logisim)을 이용해 구현함으로써 FPGA에 대한 기초 개념을 근본적으로 이해하고자 하는 데 있다. 소프트웨어에서 학습한 MLP 모델의 가중치와 편향을 정수화하여 회로에 반영하고, XOR 문제와 Iris 데이터셋 이진 분류를 직접 회로로 구현함으로써, 추상적인 인공지능 알고리즘이 실제 하드웨어에서는 어떻게 표현되고 동작하는지를 체계적으로 학습한다. 이는 향후 FPGA를 활용한 하드웨어 기반 AI 임베디드 시스템 구현을 위한 초석을 마련하는 데 중요한 역할을 한다.

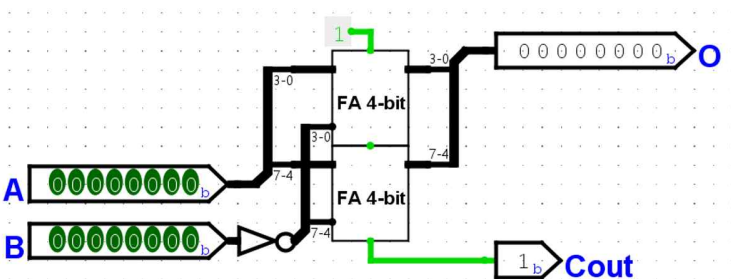
나) 이론적 배경

1) Arithmetic Operations



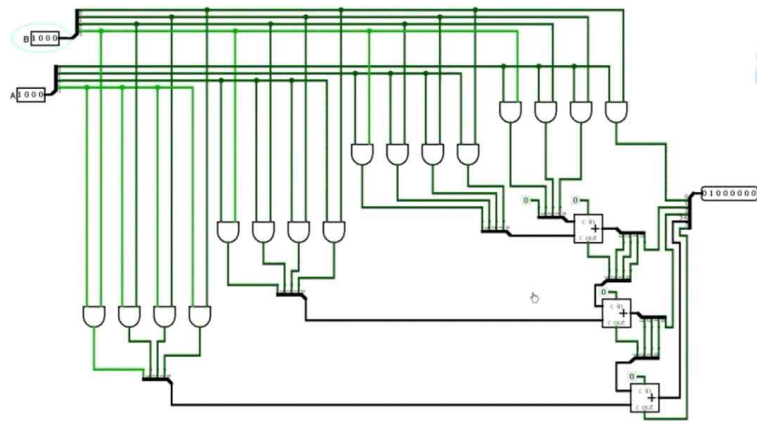
[사진 1] 가산기

8비트 가산기(8-bit adder)는 두 개의 8비트 이진수를 더해 9비트의 결과(합과 캐리)를 출력하는 디지털 회로로, 보통 1비트 전가산기(full adder)를 8개 직렬로 연결해 구성된다. 각 전가산기는 입력된 두 비트와 이전 자리에서 전달된 캐리(carry-in)를 받아 해당 자리의 합(sum)과 다음 자리로 넘길 캐리(carry-out)를 계산하며, 최상위 비트의 캐리 출력은 오버플로우 여부 판단에도 사용된다. 이 가산기는 컴퓨터의 산술 논리 장치(ALU)나 마이크로컨트롤러, 임베디드 시스템 등에서 기본 연산 유닛으로 활용된다.



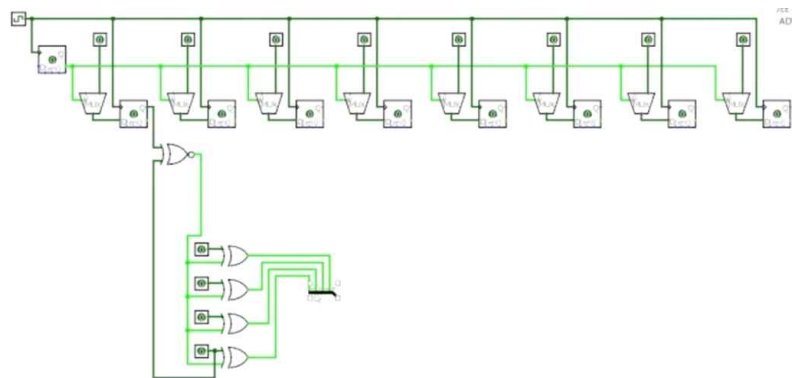
[사진 2] 감산기

8비트 감산기(8-bit subtractor)는 두 개의 8비트 이진수 A와 B를 입력으로 받아 $A - B$ 연산을 수행하고 8비트 차와 캐리 또는 버로우(borrow)를 출력하는 디지털 회로이다. 일반적으로 감산은 B의 2의 보수를 취해 A에 더하는 방식($A + (\sim B + 1)$)으로 처리되며, 이때 8비트 가산기를 재활용할 수 있다. 감산 결과가 음수가 되면 이진수 표현에 따라 2의 보수 형식으로 나타나며, 최상위 비트는 부호를 나타내는 데 사용될 수 있다. 감산기는 ALU, 디지털 신호 처리, 제어 시스템 등 다양한 디지털 회로에서 기본적인 산술 연산을 수행하는 핵심 요소 중 하나다.



[사진 3] 곱셈기

[사진 3]은 4비트 이진수 두 개를 입력받아 곱셈을 수행하고 8비트 결과를 출력하는 이진 곱셈기이다. 입력된 두 수는 AND 게이트를 통해 각 비트 단위로 곱해져 부분곱(partial product)을 생성하고, 이 부분곱들은 각 위치에 맞춰 시프트된 후 전가산기(full adder)들을 통해 누적되어 최종 곱셈 결과가 계산된다. 회로는 시프트-앤-애드(Shift-and-Add) 방식으로 구성되어 있으며, 이는 하드웨어에서 이진 곱셈을 구현할 때 가장 기본적인 방식이다. 최종 출력은 8비트로 구성되어 있어 곱셈 결과를 정확하게 표현할 수 있으며, 이 구조는 간단한 프로세서나 디지털 시스템에서 사용되는 곱셈기의 기본 형태다.



[사진 4] 나눗셈기

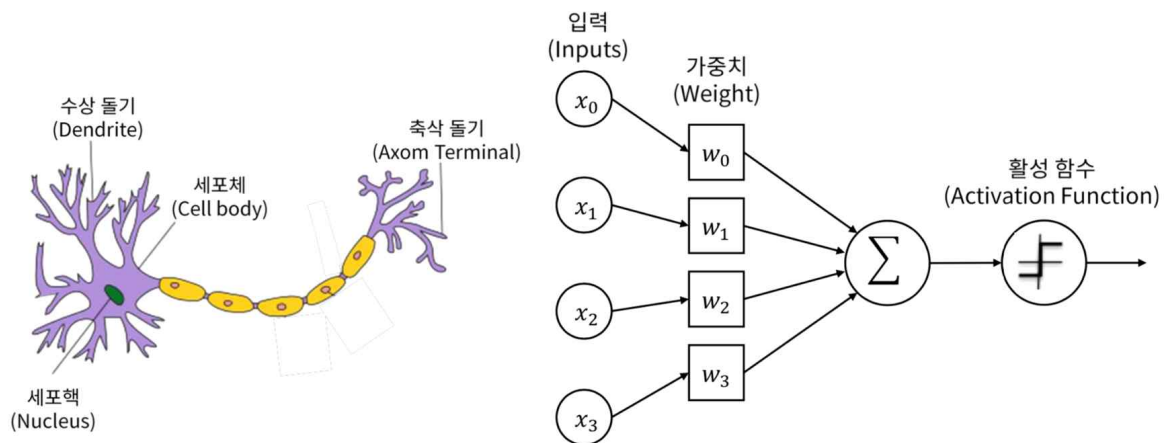
[사진 4]는 4비트 이진수를 입력으로 받아 나눗셈 연산을 수행하는 4비트 이진 나눗셈기이다. 기본적으로 피제수와 제수를 반복적으로 비교하고 뺄 뒤 시프트하면서 몫과 나머지를 구하는 방식으로 동작하며, 회로 상단에는 여러 개의 ALU 블록이 나열되어 각 단계에서 뺄셈과 시프트

트 연산을 처리한다. 비교 결과와 제어 신호들은 논리 게이트들로 구성된 하단 회로를 통해 제어되며, 이를 통해 조건 분기나 뭉 결정이 이루어진다. 이 구조는 복잡하지만 기본적인 복원 나눗셈(restoring division) 알고리즘을 하드웨어적으로 구현한 형태로, 디지털 컴퓨팅 시스템에서의 나눗셈 연산 과정을 그대로 따라간다.

II 본론

가) 퍼셉트론

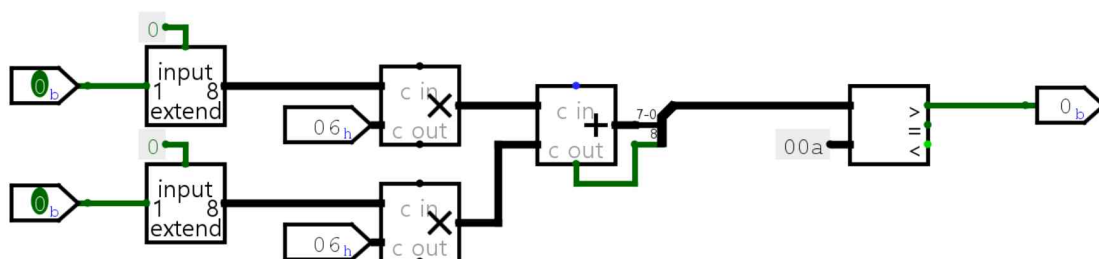
1) 퍼셉트론의 등장 배경과 구조



[사진 5] 좌 : 뉴런 구조, 우 : 퍼셉트론 구조

퍼셉트론은 1958년에 심리학자이자 컴퓨터 과학자인 프랭크 로젠블랫이 고안한 인공신경망 모델로, 인간의 시각 인식 과정을 기계적으로 구현하려는 시도에서 출발했다. 그는 퍼셉트론을 통해 컴퓨터가 학습과 분류를 수행할 수 있다는 가능성을 제시했으며, 입력값에 가중치를 곱해 모두 더한 뒤, 그 합이 특정 임계값을 넘는지 여부에 따라 이진 출력(0 또는 1)을 내는 구조로 설계했다.

2) 퍼셉트론 디지털 회로 구현

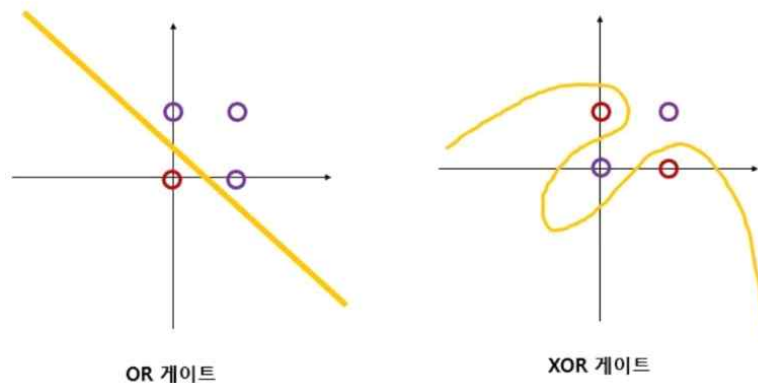


[사진 6] 퍼셉트론으로 구현한 AND 연산

[사진 6]은 두 개의 8비트 입력값에 대해 AND 연산을 수행하고, 그 결과를 더한 후 기준값인 0x0A와 비교하여 출력하는 구조이다. 입력값은 각각 0으로 설정되어 있으며, 8비트로 확장된 뒤 0x06과 곱 연산된다. 두 곱 결과를 더한 값은 비교 블록으로 전달되어, 기준값 0x0a보다 큰지(>), 같은지(=), 작는지(<)를 판단하게 된다. 현재는 입력값이 모두 0이므로 최종 연산 결과도 0이 되고, 이는 0x0a보다 작기 때문에 출력은 '<' 조건에 해당되어 0이 출력된다.

나) 다층 퍼셉트론

1) 다층 퍼셉트론의 등장 배경



[사진 7] 단층 퍼셉트론의 한계

$$\begin{aligned}
 x^{(1)} &= [0, 0], y^{(1)} = -1 \Rightarrow -b > 0 & \Rightarrow b < 0 \\
 x^{(2)} &= [0, 1], y^{(2)} = +1 \Rightarrow w_2 + b > 0 & \Rightarrow w_2 > -b \\
 x^{(3)} &= [1, 0], y^{(3)} = +1 \Rightarrow w_1 + b > 0 & \Rightarrow w_1 > -b \\
 x^{(4)} &= [1, 1], y^{(4)} = -1 \Rightarrow w_1 + w_2 + b < 0
 \end{aligned}$$

$$w_1 + w_2 > -2b$$

$$w_1 + w_2 < -b$$

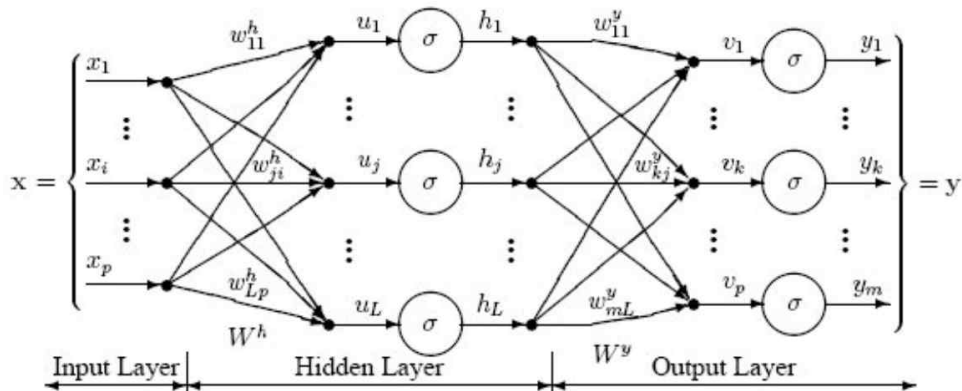
$$\text{모순} \Rightarrow \nexists (w, b)$$

XOR는 단층 퍼셉트론으로 분리 불가능

[사진 8] 선형 분리 불가능성 증명

단층 퍼셉트론(Single Layer Perceptron)은 선형적으로(Linear) 분류(Classification) 가능한 문제만 해결할 수 있다는 한계점이 있다. 이에 퍼셉트론을 여러층 쌓은 순방향의 인공 신경망, 다층 퍼셉트론(Multi Layer Perceptron, MLP)을 통해 비선형적으로(Unlinear) 분류 해야하는 문제도 해결할 수 있게 되면서 XOR 문제를 해결할 수 있게 되었다.

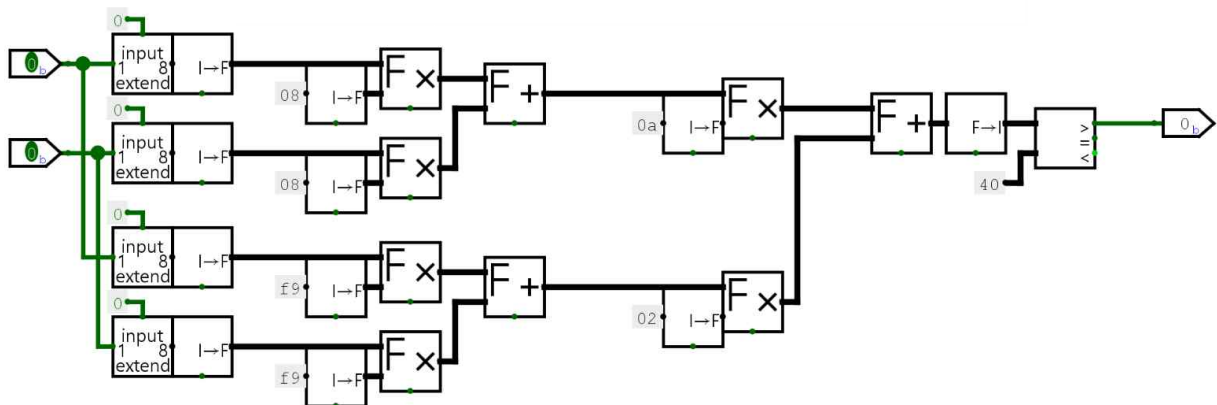
2) 다층 퍼셉트론의 구조



[사진 9] 다층 퍼셉트론 구조

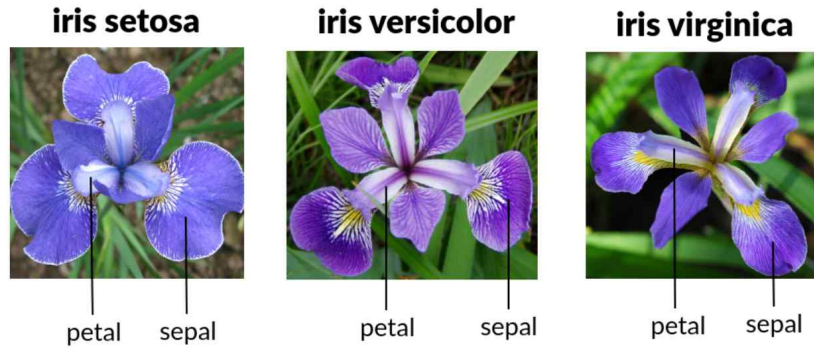
MLP는 입력층, 하나 이상의 은닉층, 그리고 출력층으로 구성된 완전 연결(feedforward) 신경망이다. 각 층의 뉴런은 이전 층의 모든 뉴런과 연결되며, 각 뉴런은 가중치 합과 편향을 바탕으로 비선형 활성화 함수를 적용해 출력을 계산한다. 은닉층이 하나만 있어도 단층 퍼셉트론보다 훨씬 복잡한 패턴을 학습할 수 있고, 층이 깊어질수록 더 높은 표현력을 갖는다.

3) 다층 퍼셉트론 디지털 회로 구현



[사진 10] 다층 퍼셉트론으로 구현한 XOR 연산

[사진 10]은 XOR 연산을 다층 퍼셉트론(Multi-Layer Perceptron) 구조로 구현한 것으로, 입력값을 부동소수점 형식으로 변환함으로써 데이터 버스 크기나 부호 처리 문제에서 자유로우며 연산의 정밀도도 확보할 수 있게 설계되어 있다. 입력값은 각기 다른 가중치(예: 0x08, 0xf9 등)와 곱해져 내적 연산을 수행하고, 이 결과들을 한 번 더 가중합 처리하여 출력층으로 전달된다. 최종 출력은 비교기를 통해 임계값 0x40과 비교되며, 이는 실제 퍼셉트론에서의 활성화 함수 역할을 한다. 회로 전체는 XOR처럼 선형 분리 불가능한 문제를 다층 구조와 비선형 처리로 해결할 수 있음을 보여주는 하드웨어적 예시이며, 각 가중치는 임계값을 기준으로 정밀하게 조정되어 정확한 출력이 도출되도록 구성되어 있다.



[사진 11] Iris 사진

Iris는 분류 예제에서 전통적으로 사용하는 Data set이다. 본 보고서에서는 다층 퍼셉트론으로 petal과 sepal을 입력값으로 받아 setosa 여부를 분류하는 예제를 수행할 것이다.

```
from sklearn.datasets import load_iris
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import numpy as np

# 1. 데이터셋 불러오기
iris = load_iris()
X = iris.data # 4개 입력 (sepal length, sepal width, petal length, petal width)
y = (iris.target == 0).astype(int) # Setosa vs Not-Setosa → 0 vs 1로 변환

# 2. 데이터 전처리 (표준화)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 3. 학습/테스트 분리
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# 4. MLP 모델 정의 및 학습
mlp = MLPClassifier(hidden_layer_sizes=(3,), activation='relu', solver='adam', max_iter=1000, random_state=42)
mlp.fit(X_train, y_train)

# 5. 정확도 출력
print("Train acc:", mlp.score(X_train, y_train))
print("Test acc:", mlp.score(X_test, y_test))

# 6. Weight, Bias 추출
print("\nWeight & Bias (원본 float 형태):")
for i, (w, b) in enumerate(zip(mlp.coefs_, mlp.intercepts_)):
    print(f"Layer {i} weights: {w}")
    print(f"Layer {i} bias: {b}")

# 7. 고정소수점 변환 (예: ×16 → 정수)
SCALE = 16

def quantize(arr):
    return np.round(arr * SCALE).astype(int)

print("\nWeight & Bias (고정소수점 스케일 ×16):")
for i, (w, b) in enumerate(zip(mlp.coefs_, mlp.intercepts_)):
    print(f"Layer {i} weights (×16): {quantize(w)}")
    print(f"Layer {i} bias (×16): {quantize(b)}")
```

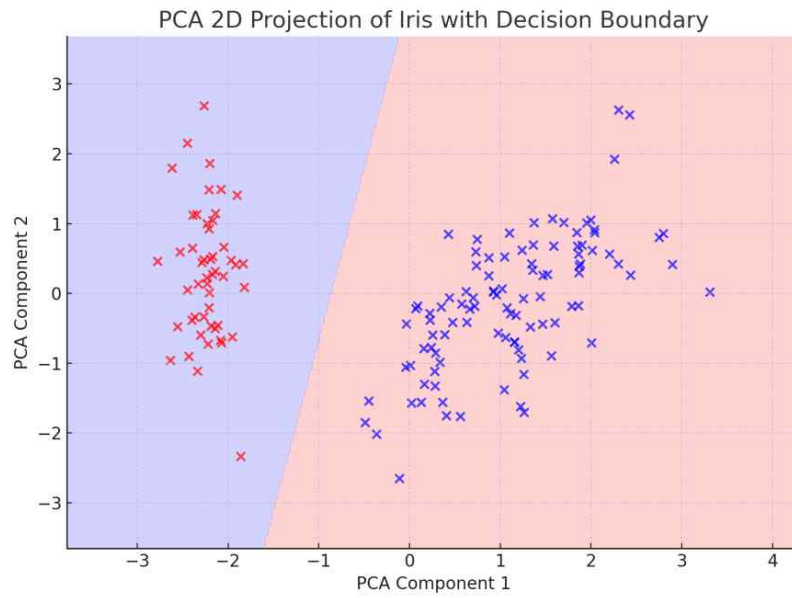
```
Train acc: 1.0
Test acc: 1.0

Weight & Bias (원본 float 형태):
Layer 0 weights:
[[ 0.22730476  1.10032093  0.40784036]
 [-0.24476985 -1.41821729 -1.63540136]
 [-0.32366186  1.31582194  1.04871552]
 [ 0.86753992 -0.20564305  1.72232564]]
Layer 0 bias:
[1.03878383 0.29682627 0.47214845]
Layer 1 weights:
[[-0.92651766]
 [-1.21915244]
 [-0.74845839]]
Layer 1 bias:
[1.34684629]

Weight & Bias (고정소수점 스케일 ×16):
Layer 0 weights (×16):
[[ 4  18  7]
 [-4 -23 -26]
 [-5  21  17]
 [ 14 -3  28]]
Layer 0 bias (×16):
[17  5  8]
Layer 1 weights (×16):
[[-15]
 [-20]
 [-12]]
Layer 1 bias (×16):
[22]
```

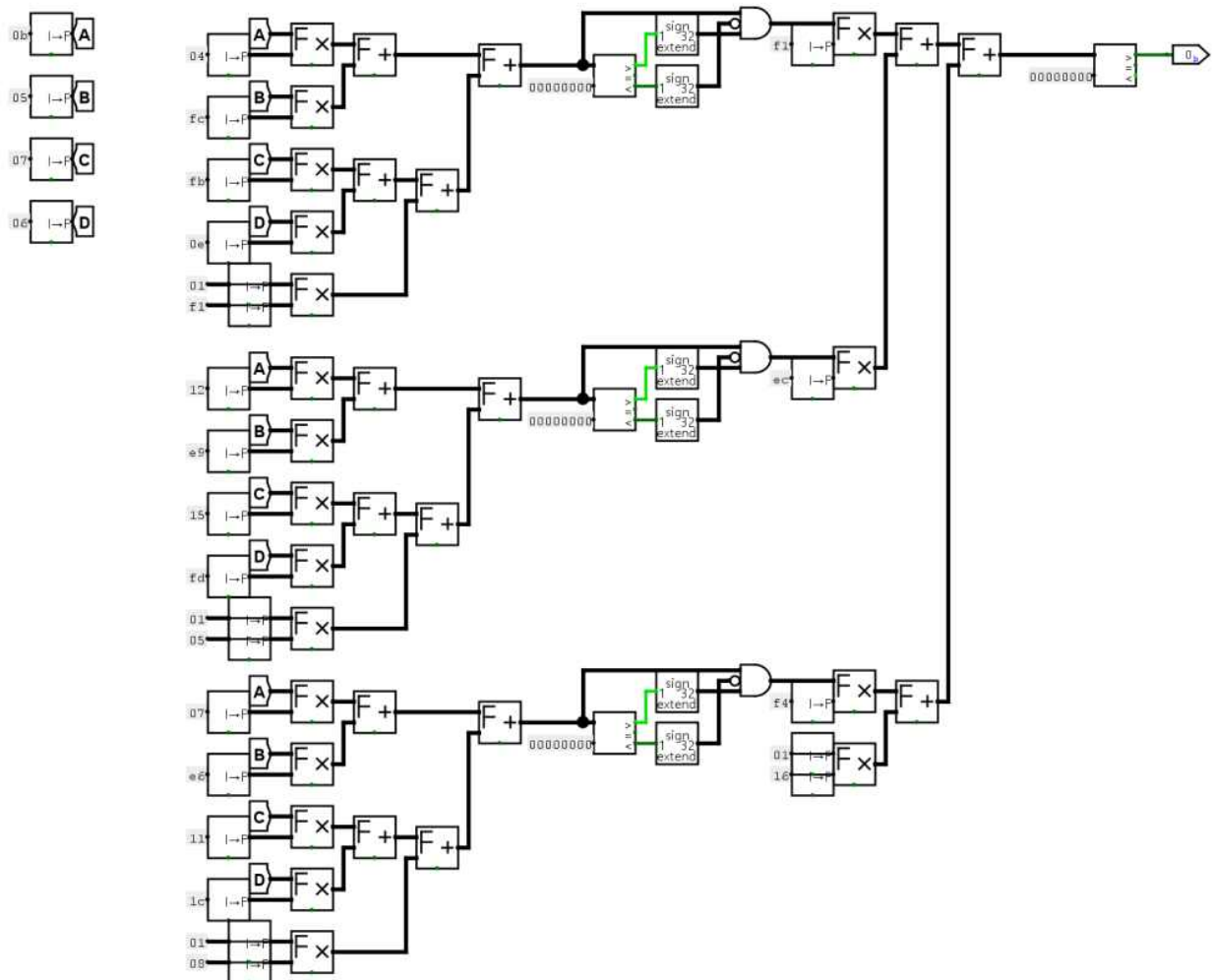
[사진 12] Iris 이진 분류 학습 코드와 결과

이 코드는 scikit-learn 라이브러리의 load_iris, StandardScaler, train_test_split, MLPClassifier를 이용해 Iris 데이터셋을 불러오고, Setosa 품종을 이진 분류하는 다층 퍼셉트론(MLP) 모델을 학습한 뒤, 학습된 가중치(weight)와 편향(bias)을 numpy를 사용해 고정소수점 스케일(×16)로 정수 변환하여 출력함으로써 회로 구현에 바로 활용할 수 있도록 처리하는 전체 파이프라인을 구성한다.



[사진 13] 주성분 분석을 통한 결정 경계 시각화

주성분 분석을 통해 분석한 결과 setosa와 그 외가 명확히 구분되는 것을 확인할 수 있었다.



[사진 14] Iris 이진 분류 다층 퍼셉트론 디지털 회로

앞에서 이진 분류 학습을 통해 얻은 가중치를 바탕으로 단층 퍼셉트론을 여러 개 붙혀 세토사와 그 외의 꽃을 구별하는 분류모델을 회로로 구현하였다. 총 4개의 입력이 있으며 각각 sepal length, sepal width, petal length, petal width 이다. 이 입력에 따라 1이면 세토사이고, 아니라면 그외의 다른 종류의 꽃이다. 데이터 크기와 임계값 연산을 쉽게 하기 위해 부동소수점 연산 모듈을 사용하였고, 가중치값과 내적 연산을 반복하는 회로를 구성하여 우리가 원하는 출력값을 얻을 수 있었다.

III 결론

본 연구는 인공지능 알고리즘을 단순한 소프트웨어 개념에서 벗어나 디지털 회로로 구현함으로써, 추상적인 AI 구조를 실제 하드웨어 수준으로 구체화할 수 있음을 확인하였다. 단층 퍼셉트론의 한계를 극복한 다층 퍼셉트론 구조를 바탕으로 XOR 문제나 Iris 이진 분류 문제 같은 선형적으로 분리 불가능한 문제도 회로로 해결할 수 있었고, 학습된 모델의 가중치를 회로에 적용하는 기술적 흐름 역시 효과적으로 구축되었다. 이러한 작업은 FPGA 기반의 하드웨어 AI 구현에 필요한 기초 역량을 갖추는 데 중요한 출발점이 되며, 향후 물리적 AI 시스템 개발에 실질적인 기여를 할 수 있다.

IV 참고 문헌

- 1) Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), 386 - 408.
- 2) Hecht-Nielsen, R. (1992). Theory of the Backpropagation Neural Network. *Neural Networks for Perception*.
- 3) scikit-learn documentation: <https://scikit-learn.org/>
- 4) Logisim digital circuit simulator: <http://www.cburch.com/logisim/>
- 5) Iris 데이터셋 - Fisher, R.A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2), 179 - 188.