

# Toward a Hardware Implementation of Lidar-based Real-time Insect Detection

SPIE Real-time Imaging and Deep Learning 2022

Trevor C. Vannoy, Elizabeth M. Rehbein, Riley D. Logan, Joseph A. Shaw, Bradley M. Whitaker



This material (AFRL-2022-1298) is based on research sponsored by Air Force Research Lab under agreement number FA8650-21-1-1176.

# Monitoring insects

# Why monitor insects? Population decline



Photo by Sara Kurféß on Unsplash

# Why monitor insects? Ecosystem damage



Connecticut Agricultural Experiment Station , Connecticut Agricultural Experiment Station, Bugwood.org  
licensed under a Creative Commons Attribution-Noncommercial 3.0 License.

# Traditional methods are slow



USDA Forest Service - Cœur d'Alene Field Office, Bugwood.org  
licensed under a Creative Commons Attribution-Noncommercial 3.0 License.

# Traditional methods are slow



Gary L. Piper, Washington State University, Bugwood.org  
licensed under a Creative Commons Attribution 3.0 License.

# Traditional methods are slow

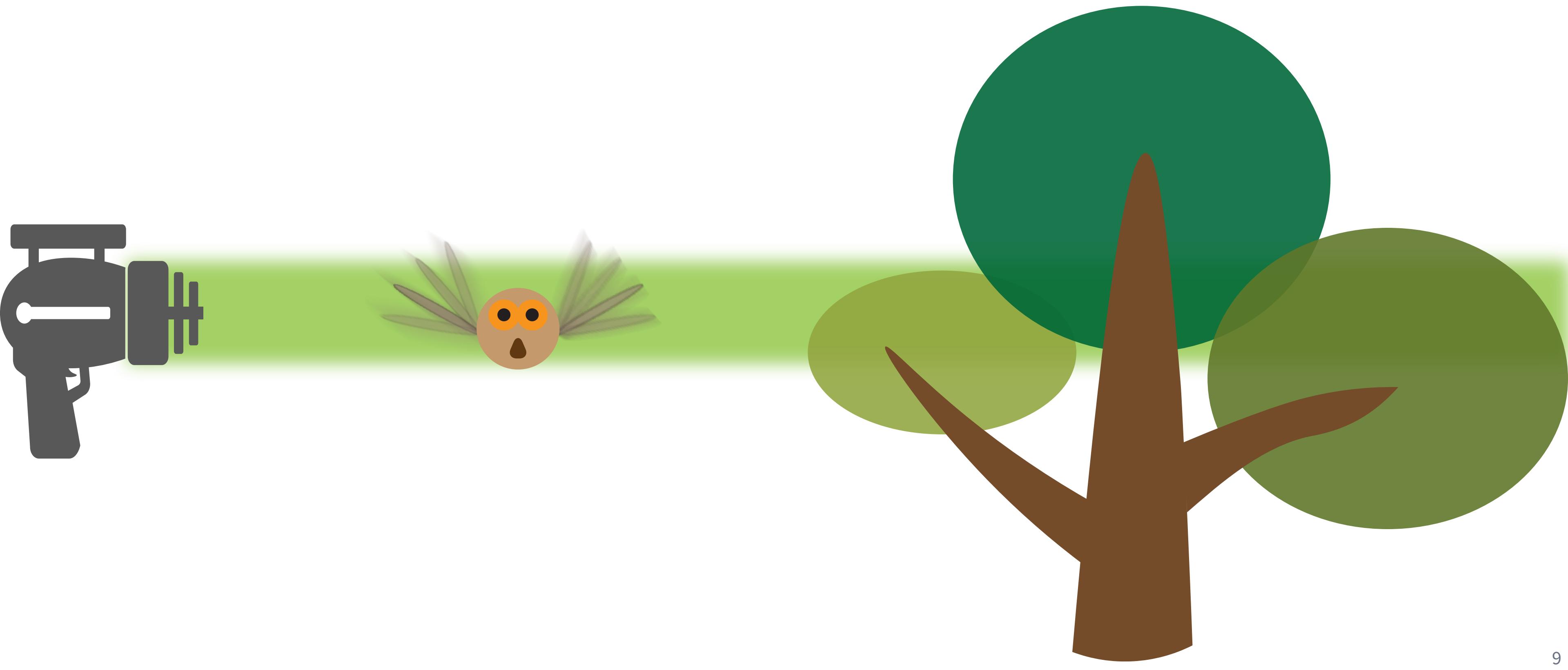


Norman E. Rees, USDA Agricultural Research Service - Retired, Bugwood.org  
licensed under a Creative Commons Attribution 3.0 License.

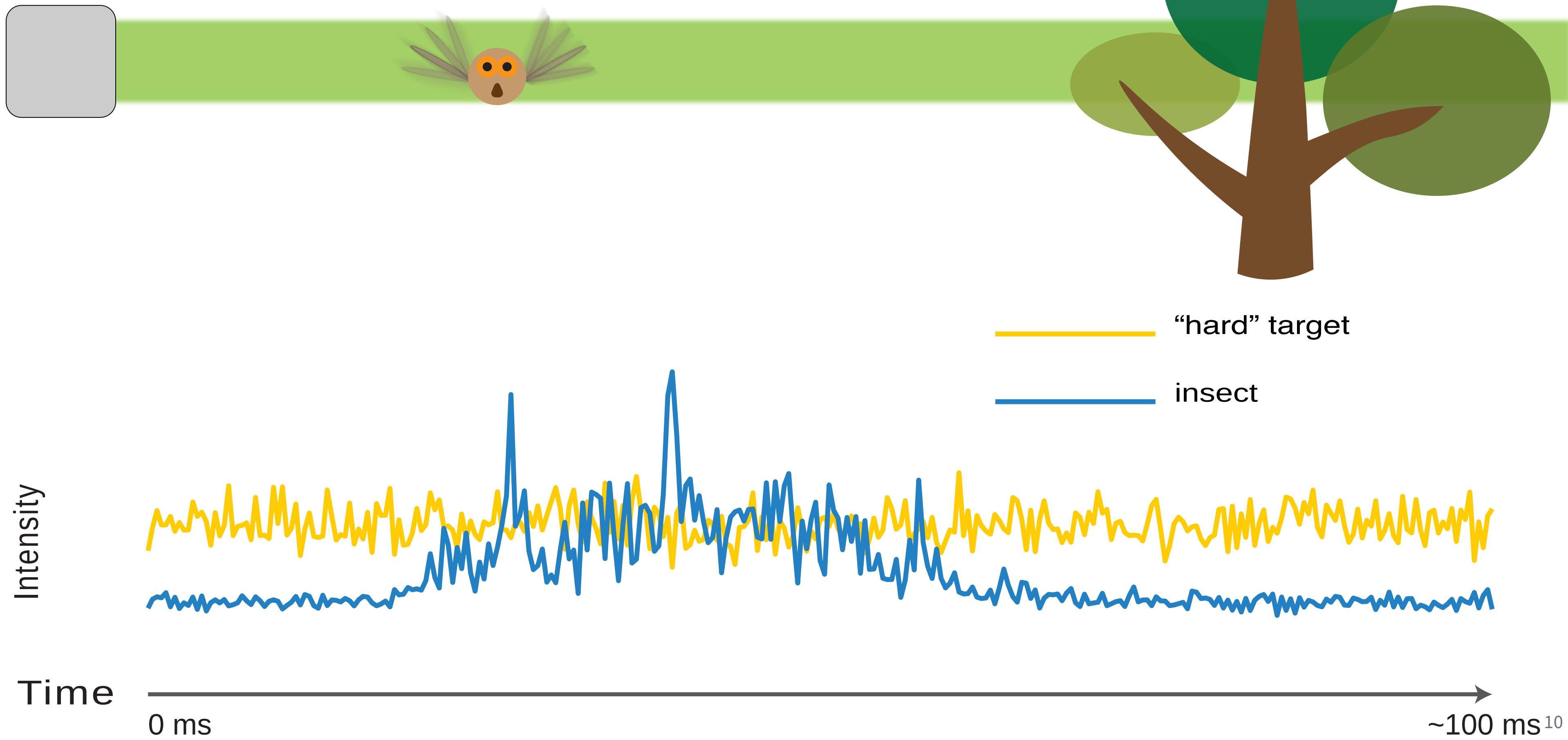
# Monitoring insects with lidar



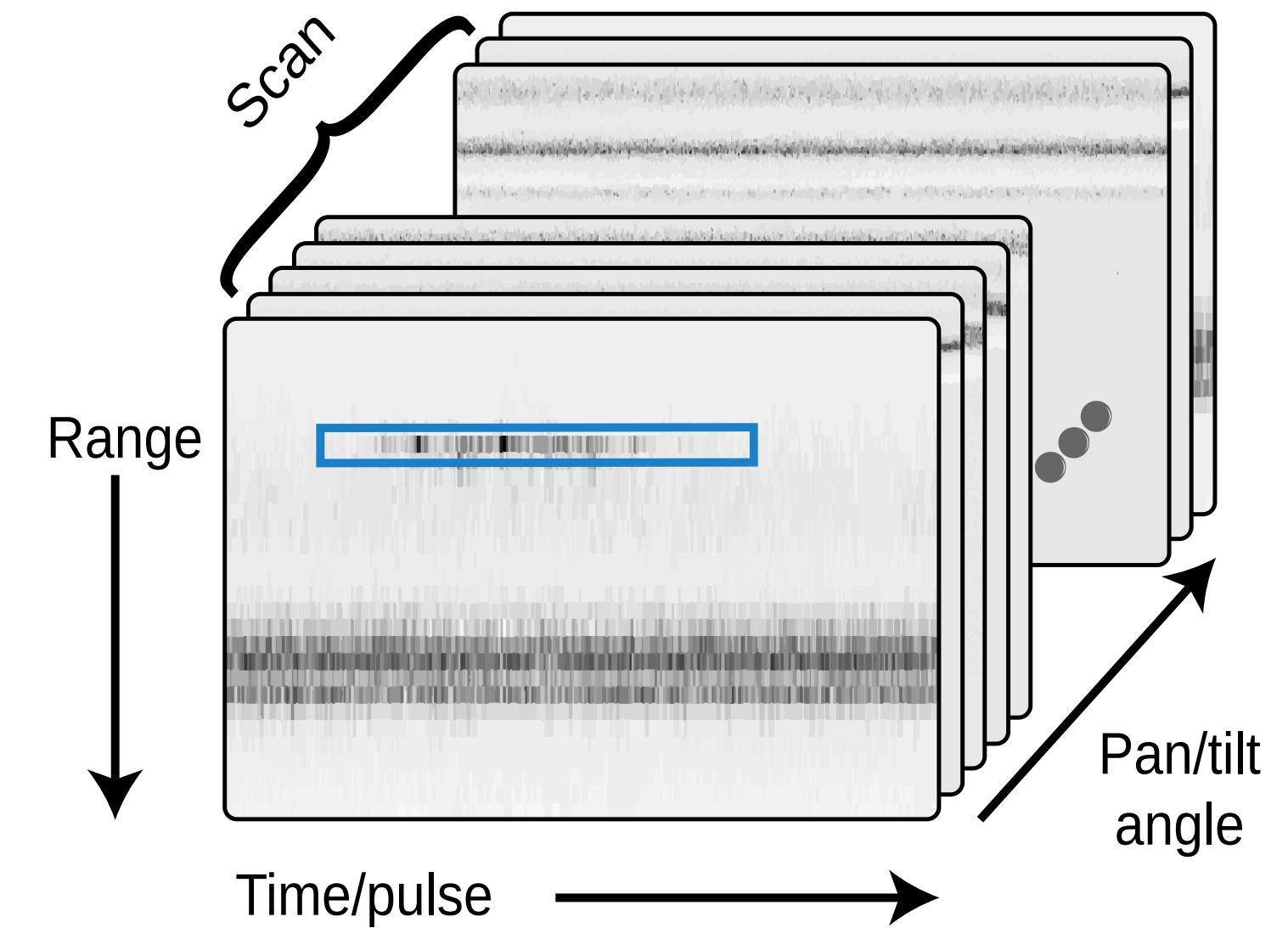
# Monitoring insects with lidar



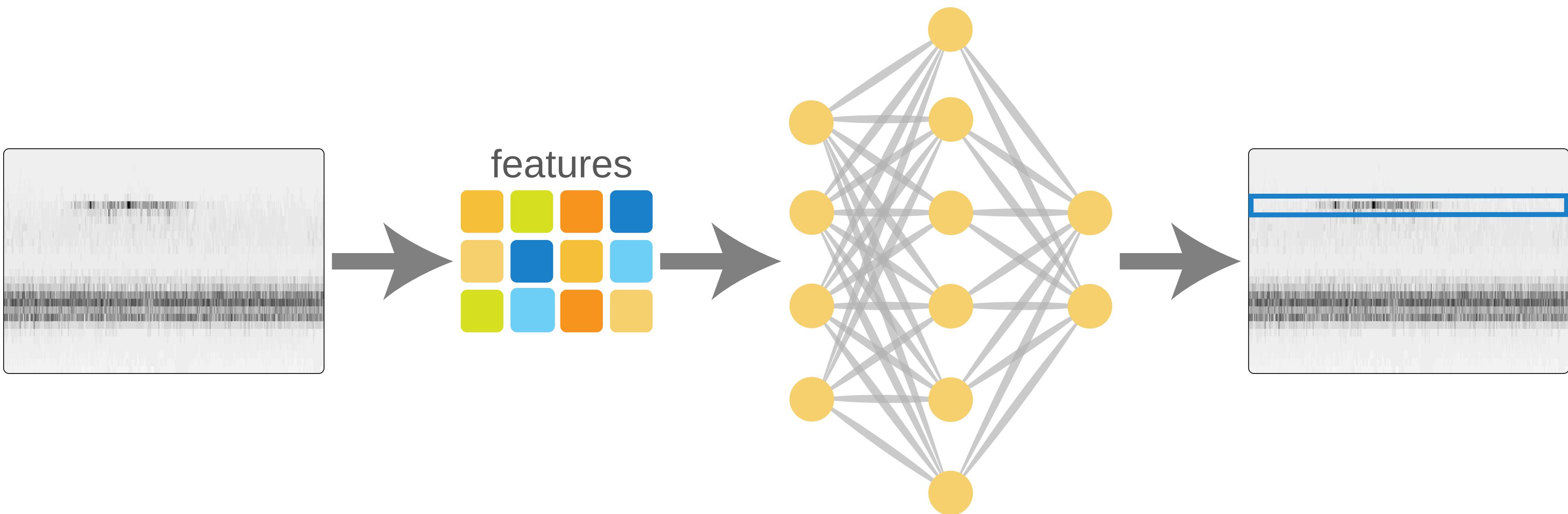
# Insect detection intuition



# Lidar system and data

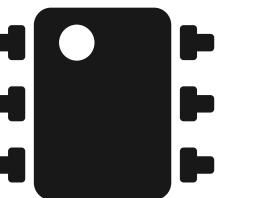


# Insect detection with a neural network

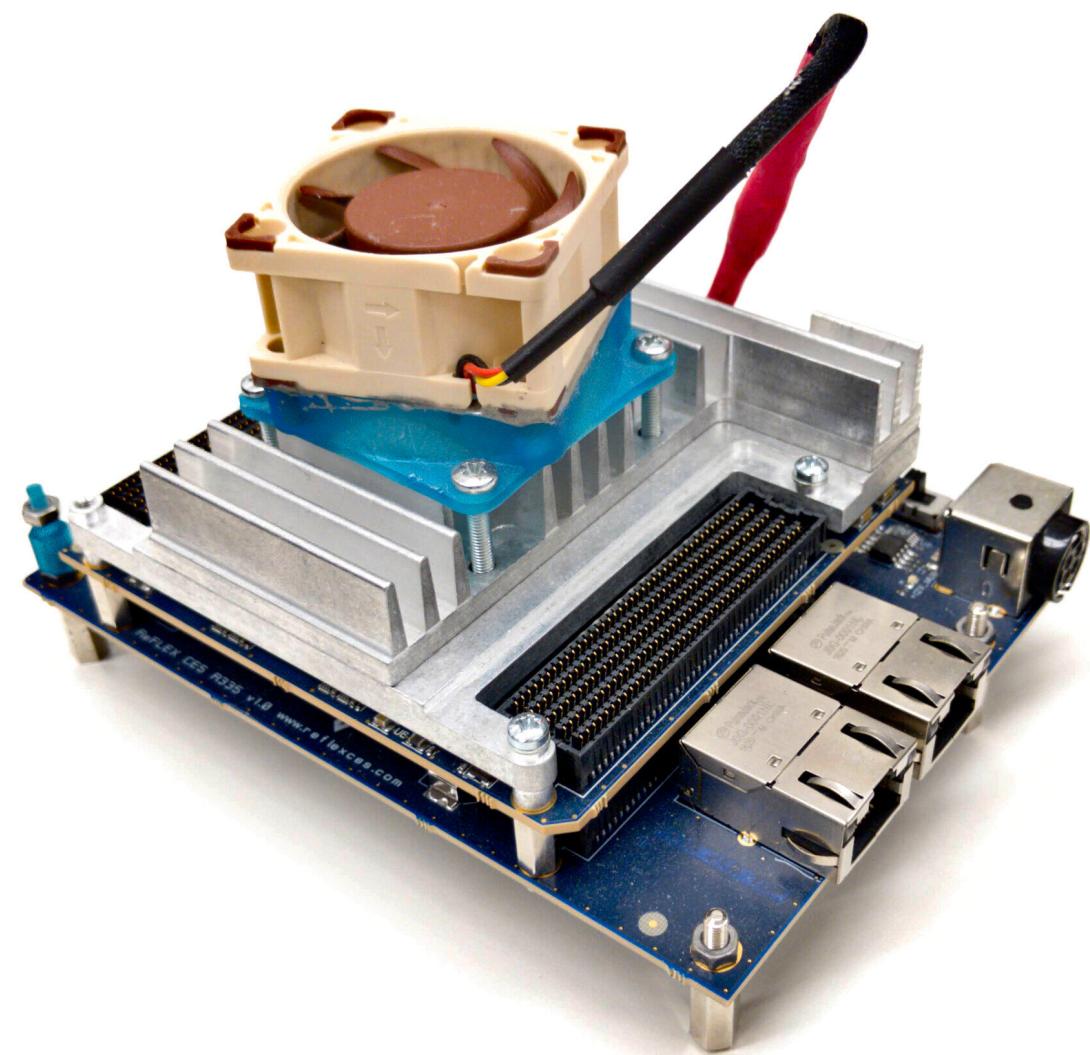


# Previous work limitations

- Offline analysis
- Not embedded



**Goal:**  
**implement insect detection on an**  
**FPGA**

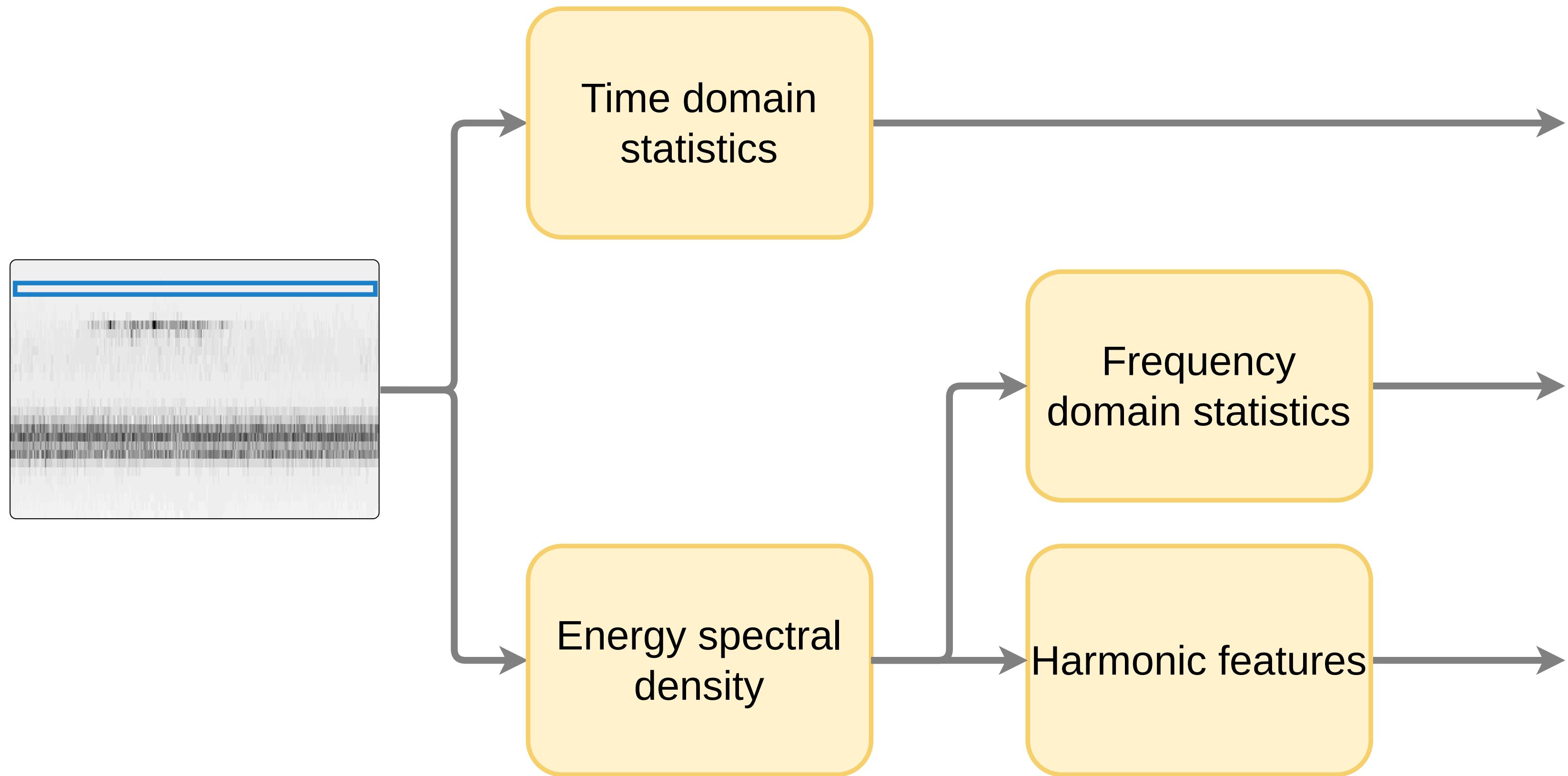


# What needs to be done

- Convert feature extraction software into HDL
- Convert neural network inference software into HDL

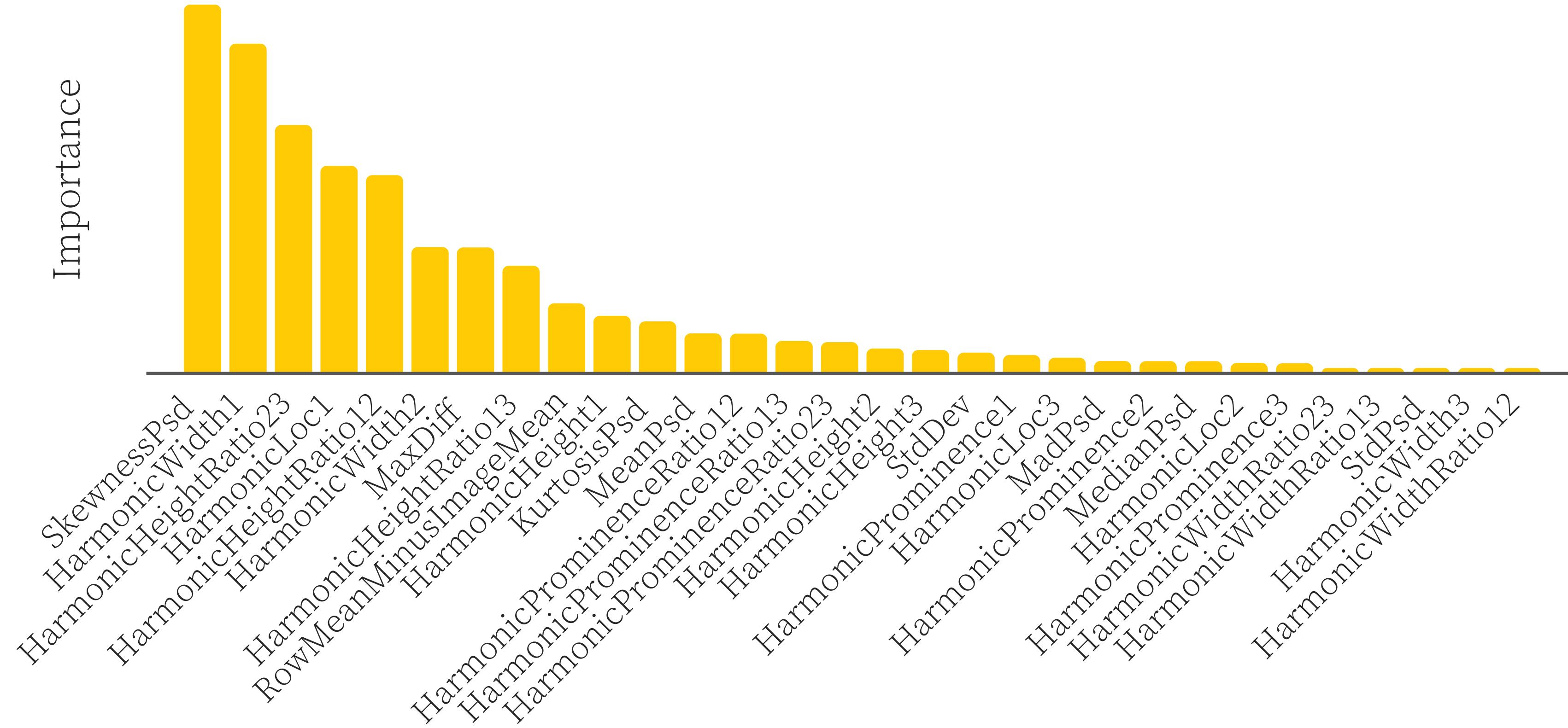
# Feature extraction overview

# Feature extraction process



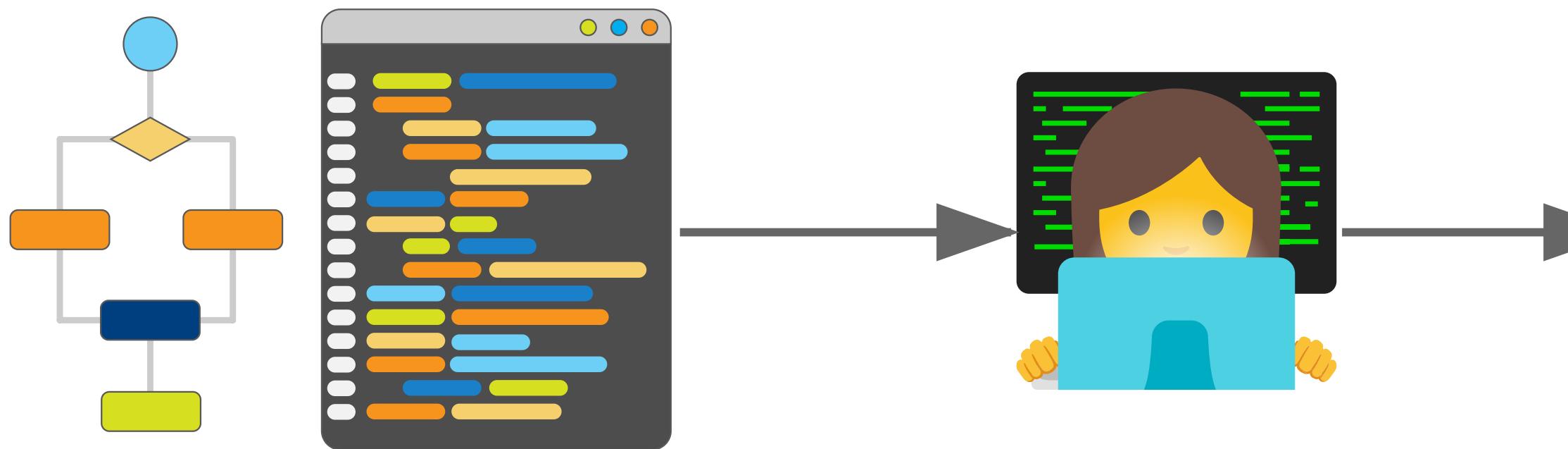
# Features listed by importance

(Minimum Redundancy — Maximum-Relevance)

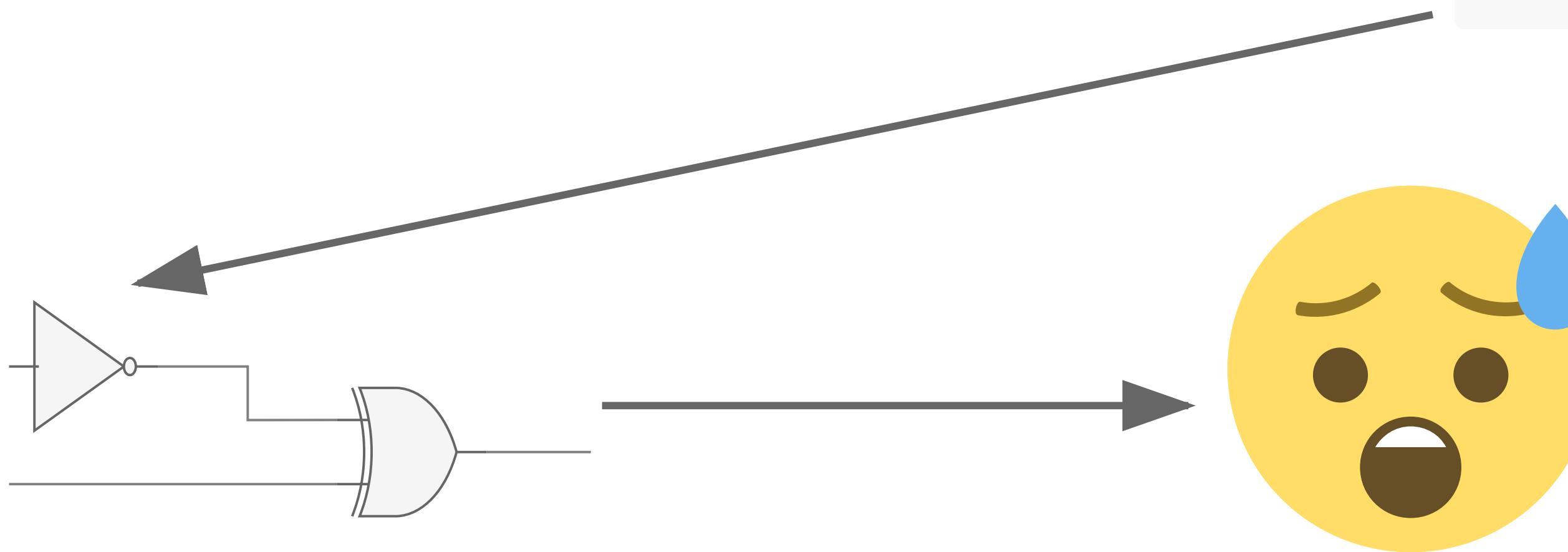


# Software-to-hardware conversion

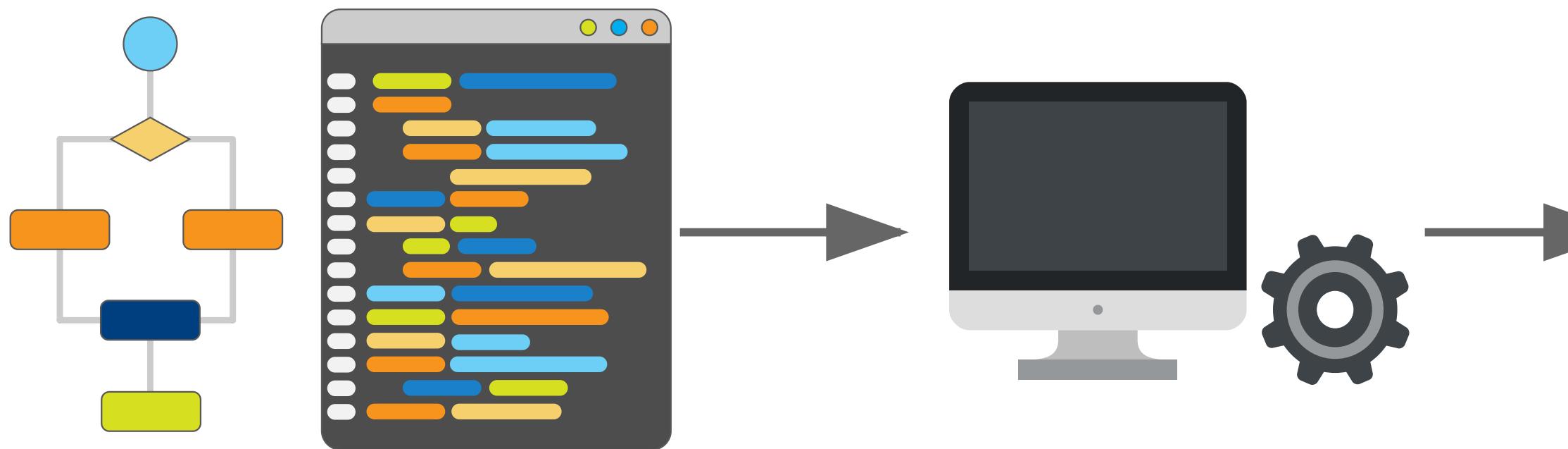
# Algorithms → hardware: the traditional way



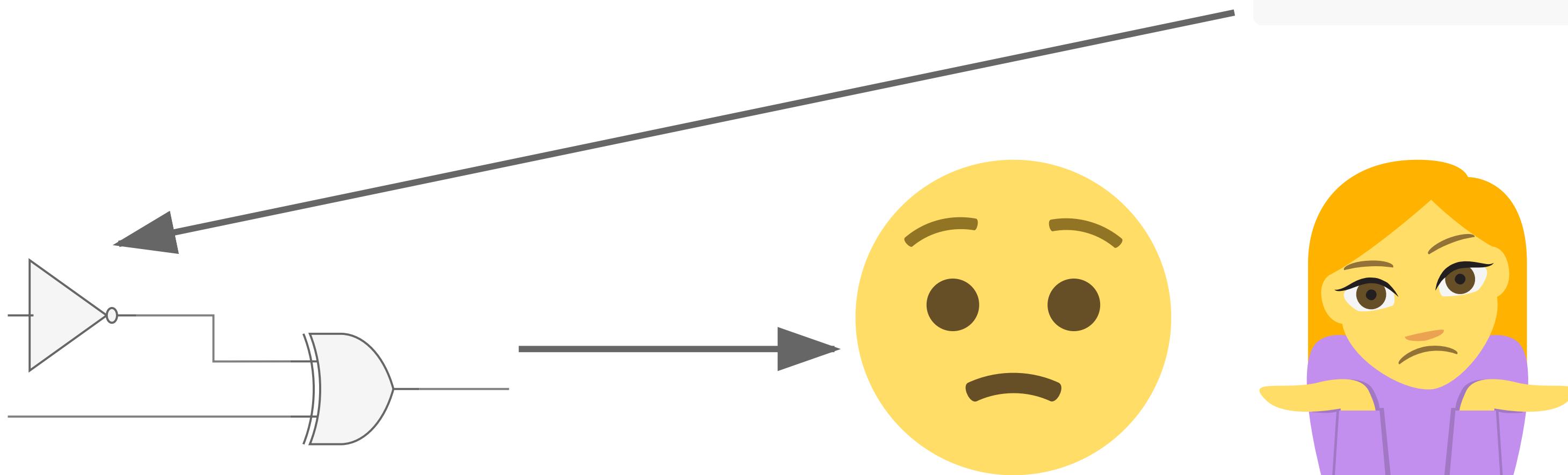
```
entity presentation is
  port(
    clk : in std_logic;
    rst : in std_logic;
    d   : out unsigned(7 downto 0)
  );
end entity;
...
a <= not b xor c;
d <= left_shift(var(7 downto 0),2);
...
```



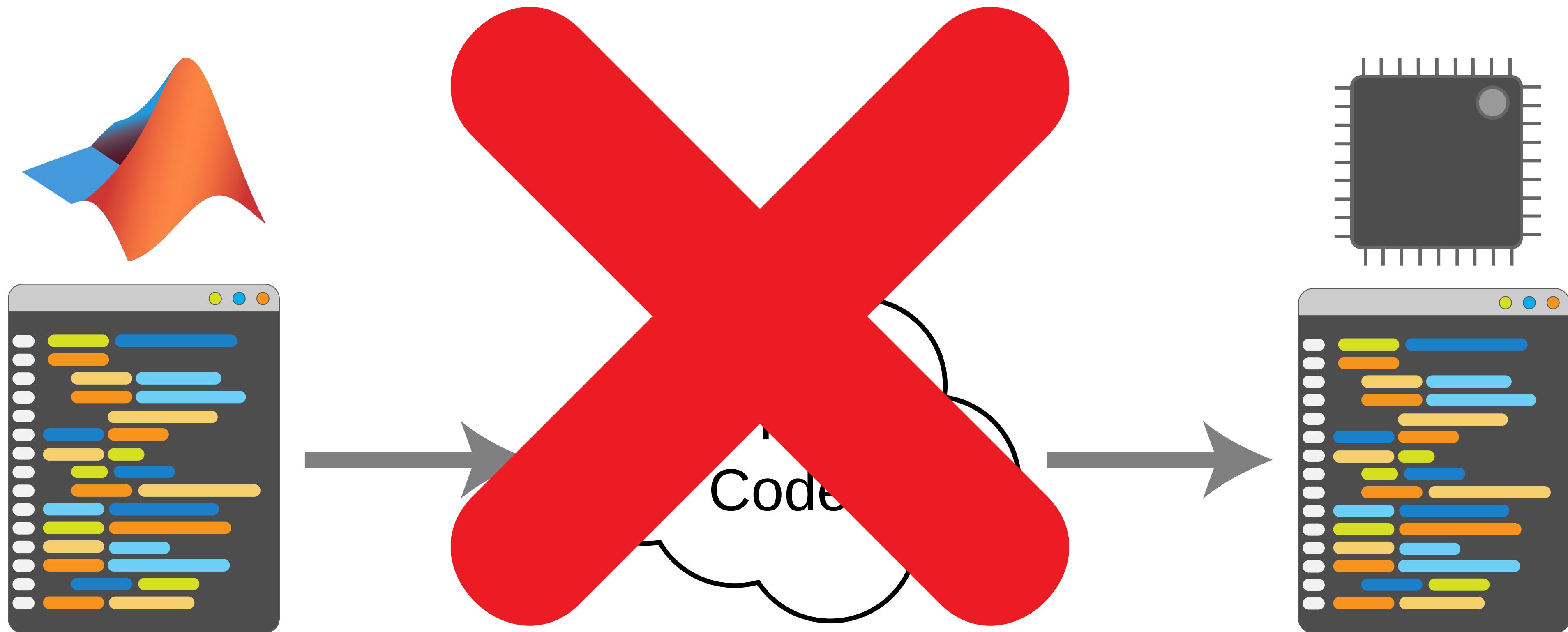
# Algorithms → hardware: using high-level synthesis



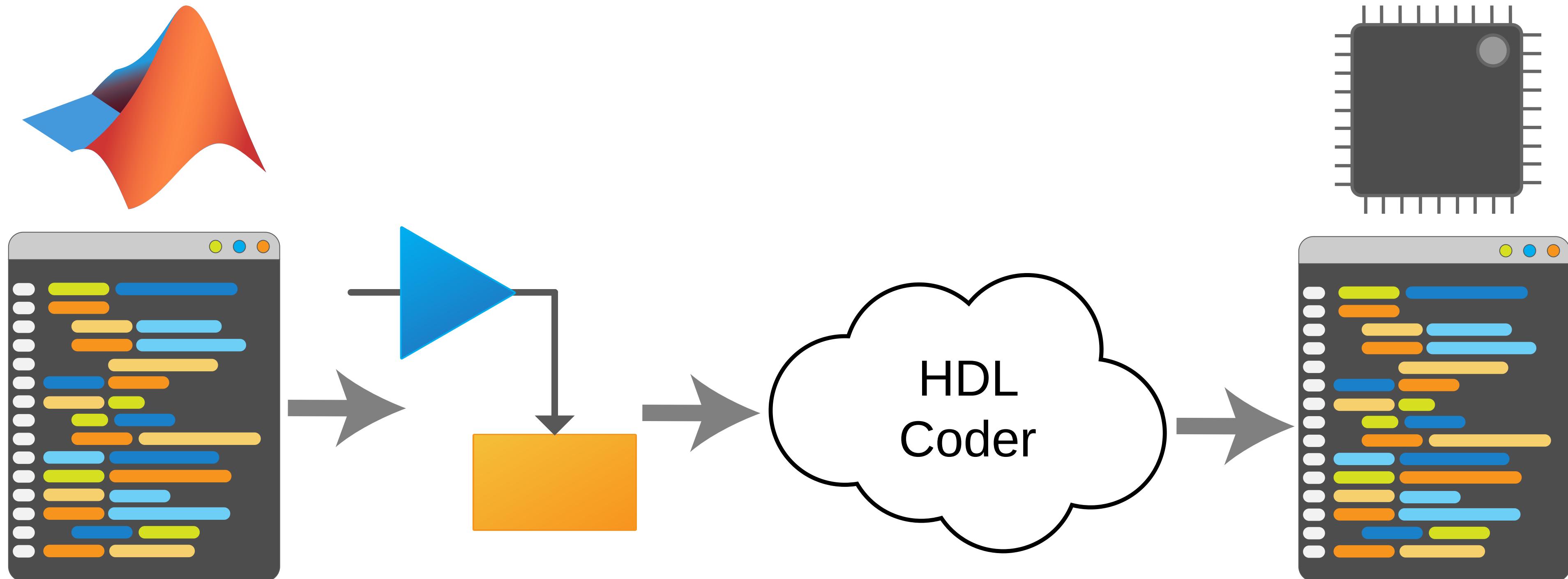
```
entity presentation is
  port(
    clk : in std_logic;
    rst : in std_logic;
    d   : out unsigned(7 downto 0)
  );
end entity;
...
a <= not b xor c;
d <= left_shift(var(7 downto 0),2);
...
```



# Convert MATLAB to HDL



# Use Simulink instead



# Implementation challenges

- HDL Coder has poor error traceability
- Median isn't easy to implement in a hardware-friendly way
- Peak finding algorithm was not hardware-friendly
  - Used *2.8 million* adders
- Features had a large dynamic range
- Design used *way too many* resources

# Future directions

- Increase resource sharing
- Implement a hardware-friendly peak finding algorithm
- Try using a DSP chip instead of an FPGA

## Takeaway:

FPGAs are still hard to develop for, even with high-level synthesis tools

# Backup slides

# Lidar system diagram

