Finger Finder: A Low-Energy Peak Detection Accelerator for Capacitive Touch Controllers

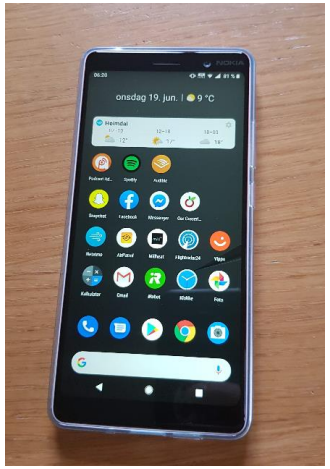
Third Workshop on Computer Architecture Research with RISC-V (CARRV 2019)
Saturday June 22nd, 2019

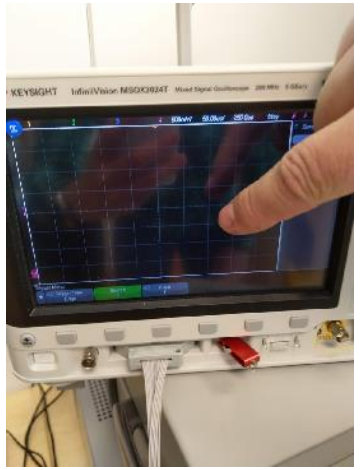
Kai Kristian Amundsen (MyWo AS), Gaute Myklebust (MyWo AS),
Per Gunnar Kjeldsberg (NTNU) and Magnus Jahre (NTNU)

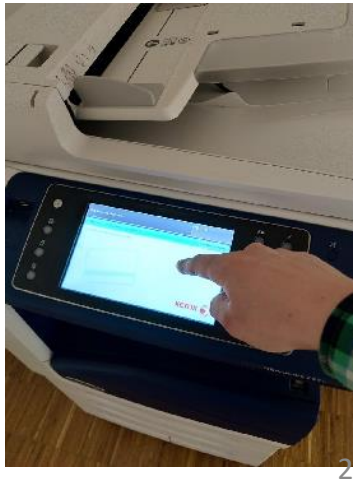
Touch screens everywhere

Home

Work

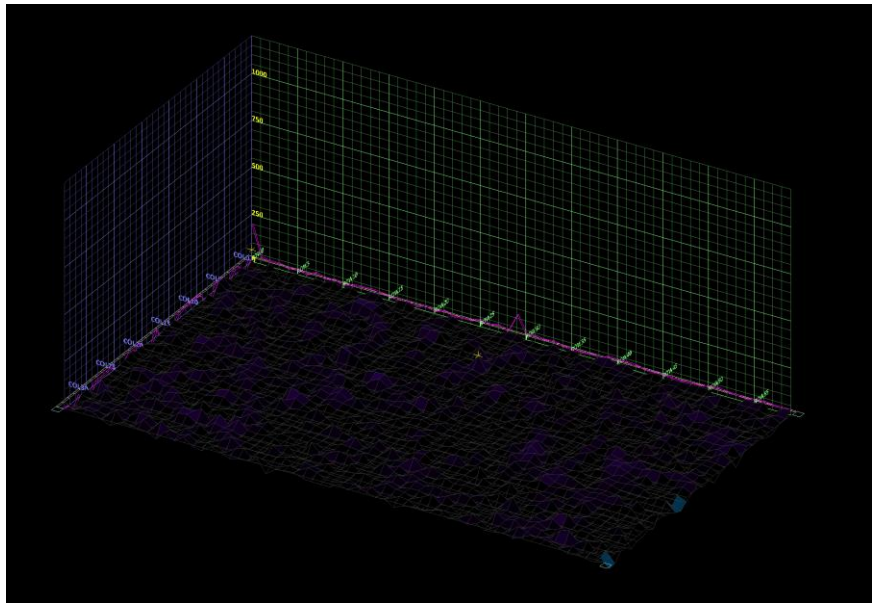




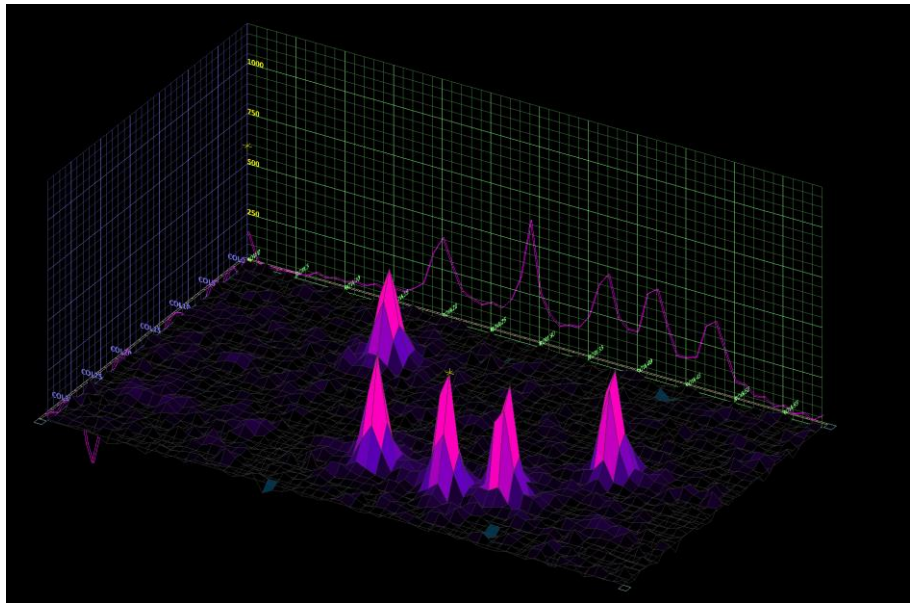


Touch screen controllers, process touch or sleep

- Peak detection is the first algorithm run on the acquired touch screen image



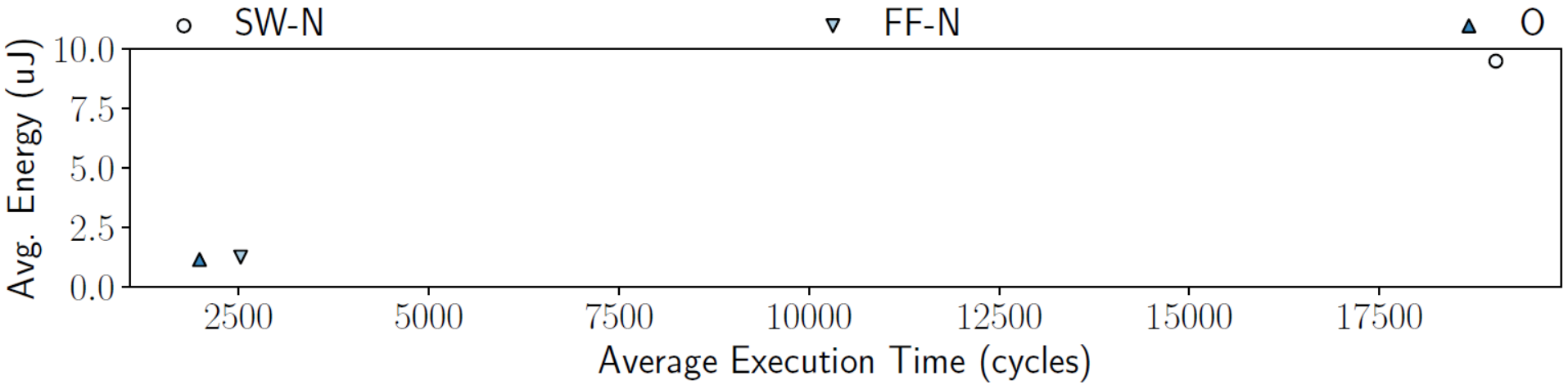
Go to sleep



Process touch

Touch screens, Finger Finder

- Finger finder (FF-N) HW accelerator
  - Compared to the best compiled C-code solution, FF-N reduces the energy consumption with 86%
  - 9.5% above a theoretical lower bound for energy consumption

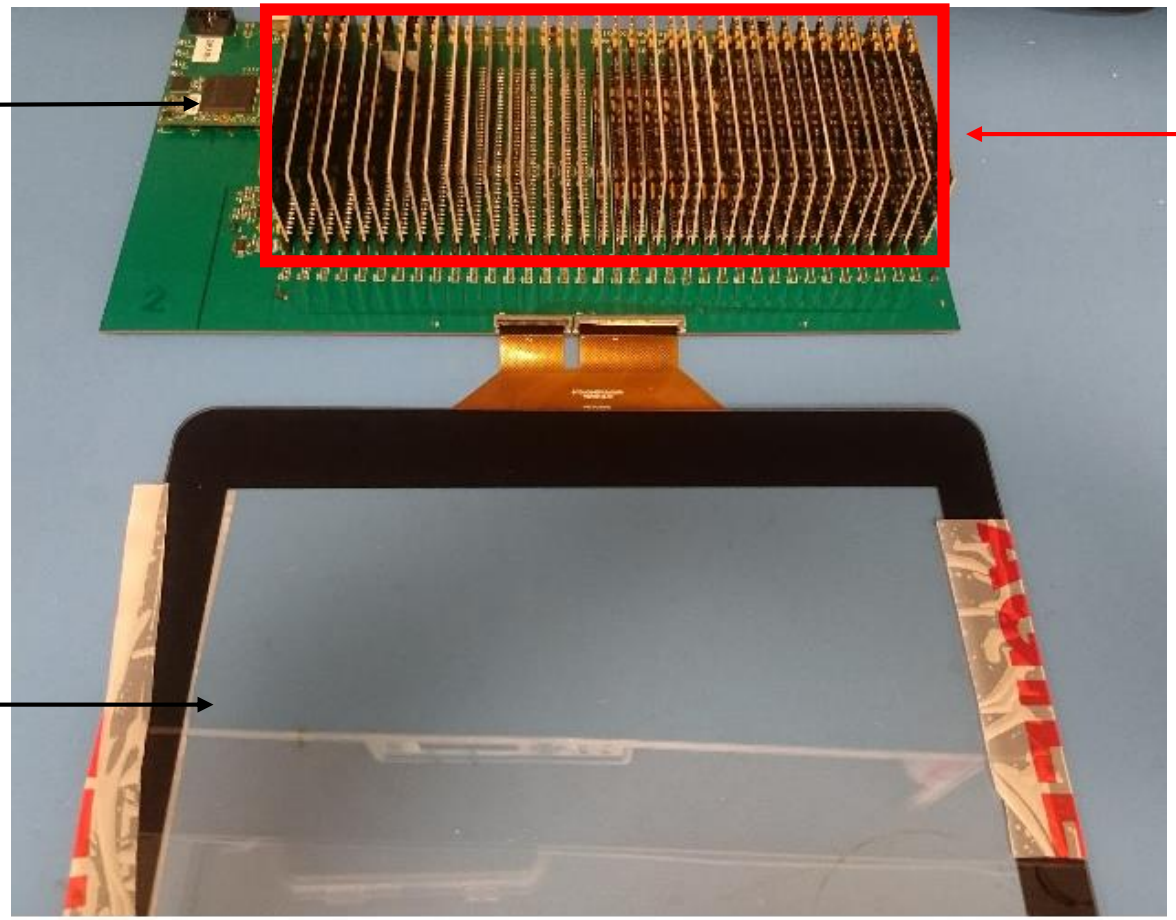


Outline

- **Introduction**
- Peak detection in SW
- Finger Finder

Why make a touch screen controller ASIC?

FPGA with CPU, IO and custom DSP



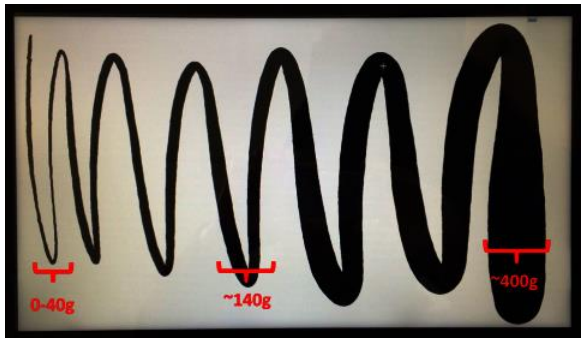
Analog front-end daughter cards

Touch screen

aXiom touch screen controller

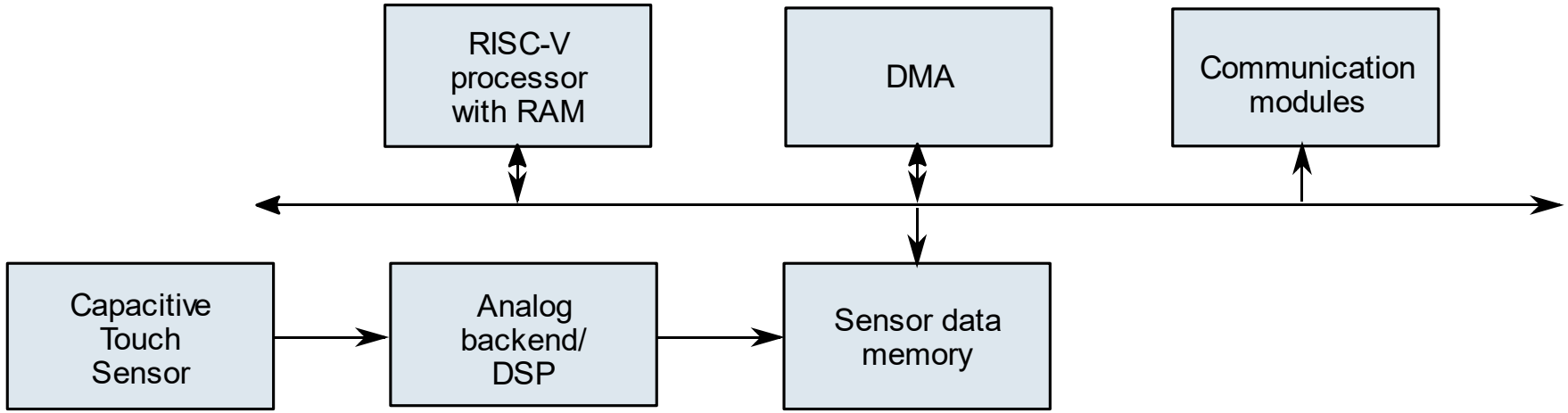
MyWo AS - designing high performance touch controllers

- Targeting industrial and automotive applications
  - Noisy environments require higher SnR
  - 3D touch
  - Force sensing
  - Haptics feedback
- Need a 32-bit CPU with mul+div



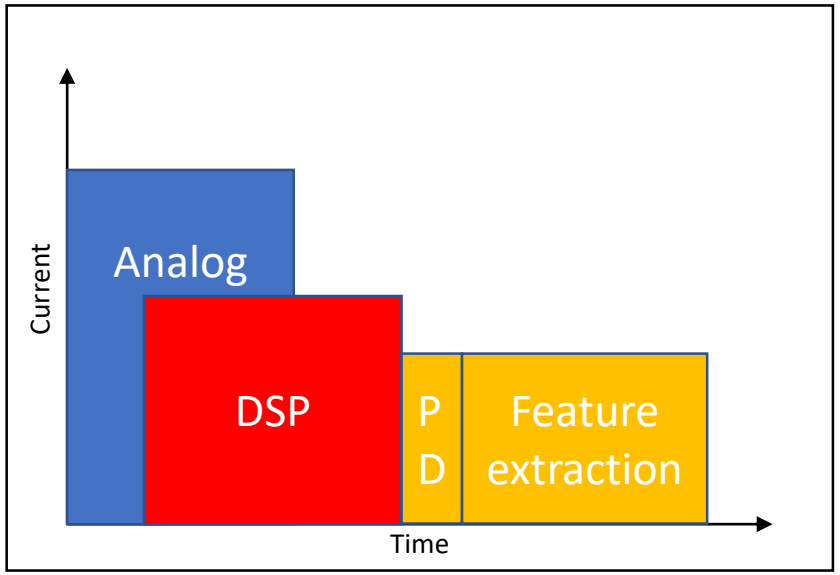
Why using RISC-V in the MyWo touch screen controller?

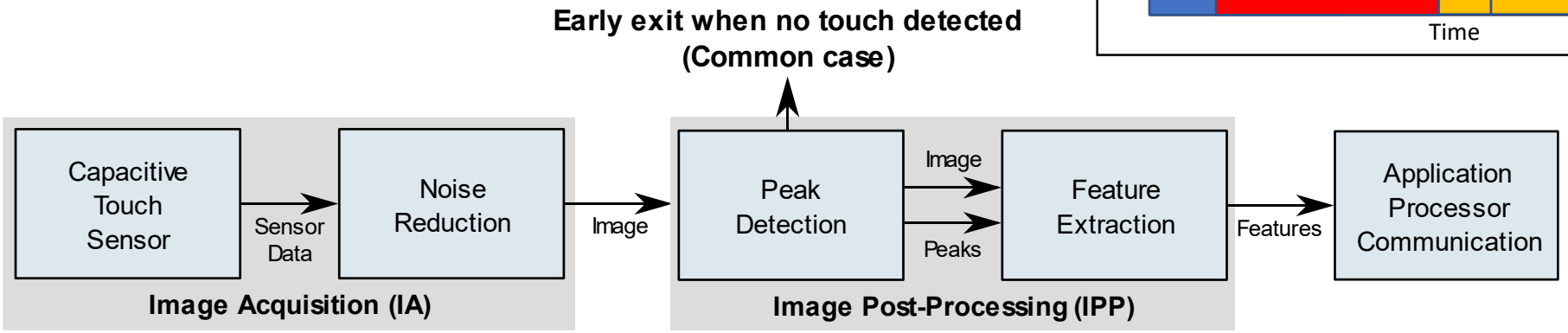
- Using the Rocket Chip generator
  - Speed
  - Cost
  - Flexibility
  - Design time



Reducing power consumption in touch controllers

- Periodically scans the touch screen
- Analog dominates first
- Then digital when processing
- Go to sleep quickly





Why peak detection?

- Selected when we had the FPGA prototype
  - The feature extraction was not complete
  - Peak detection dominated the image post-processing
    - The CPU ran at half of the speed of the aXiom chip
  - The analog current consumption was not known
- Many devices run the touch controller when the screen is off
  - Knock to wake up
  - Gestures to perform functions while off

Outline

- Introduction
- **Peak detection in SW**
- Finger Finder

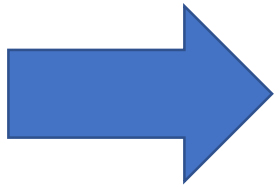
Peak detection

- Detect local maxima in a 2D image
  - Above a noise threshold
  - Peak if center is greater or equal to neighbors

1	4	2	1
5	9	8	4
3	7	7	6
2	3	5	8

- If one or more of the neighbor nodes are equal to the center peak, increment the center
  - Avoiding a large number of peaks for plateaus

1	4	2
5	9	8
3	7	9



1	4	2
5	10	8
3	7	9

Peak detection algorithms, SW-N

- The naive implementation:

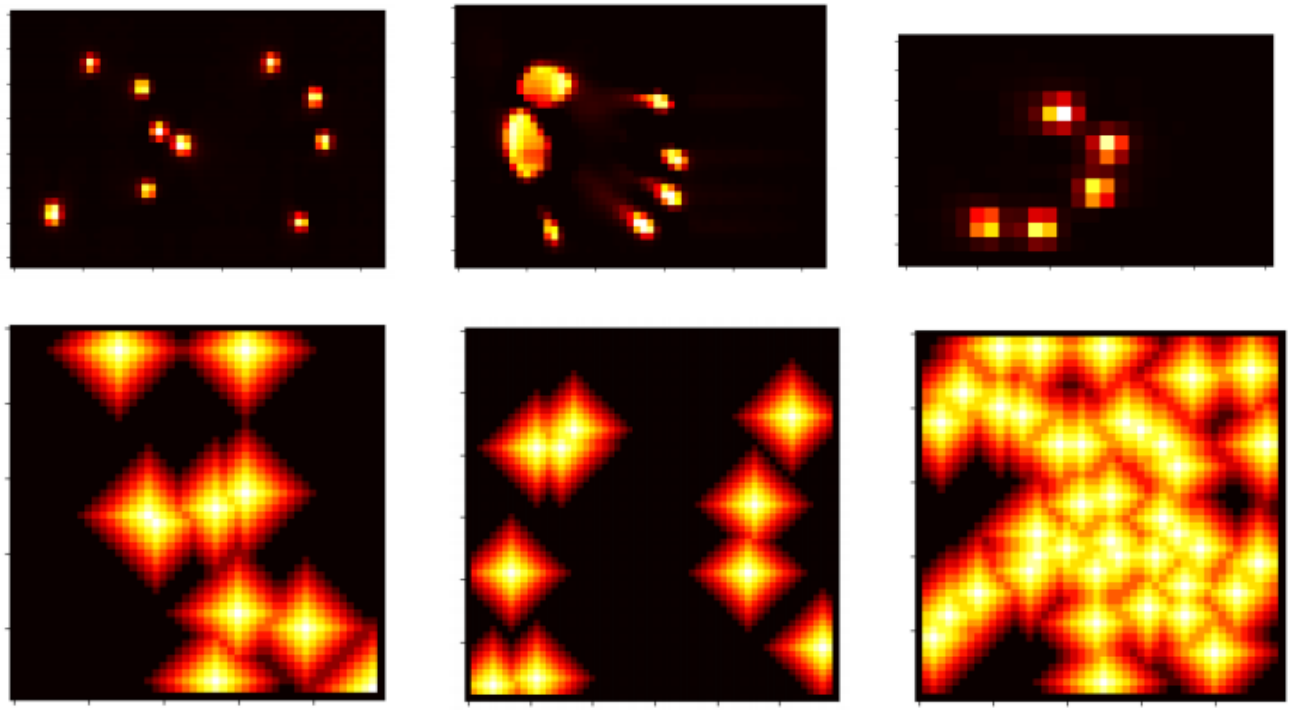
```
for (Down = DownOffset; (Down <= (Max_Down -1)) && (TotalPeaksFound < MaxPeaks); Down++)
{
    for (Across = AcrossOffset; Across <= (Max_Across -1); Across++)
    {
      if ((RawData[Down][Across] >= MinimumSignal) &&
          (RawData[Down+0][Across-1] <= RawData[Down][Across]) &&
          (RawData[Down+0][Across+1] <= RawData[Down][Across]) &&
          (RawData[Down-1][Across+0] <= RawData[Down][Across]) &&
          (RawData[Down+1][Across+0] <= RawData[Down][Across]) &&
          (RawData[Down-1][Across-1] <= RawData[Down][Across]) &&
          (RawData[Down-1][Across+1] <= RawData[Down][Across]) &&
          (RawData[Down+1][Across-1] <= RawData[Down][Across]) &&
          (RawData[Down+1][Across+1] <= RawData[Down][Across]) )
        {
```

Peak detection algorithms, SW-C, SW-N16, SW-T

- Three other software algorithms were developed:
  - SW-C will cache the 8 neighbor nodes to reduce the number of memory loads
  - SW-T will only keep the cache updated if the current node is over the threshold
  - SW-N16 tries to take advantage of the 16-bit values stored in a 32-bit RAM and reads two and two values for each load
    - Otherwise similar to SW-N
- Theoretical lower bound
  - O (Oracle) loads a node from RAM and knows if it is a peak or not.

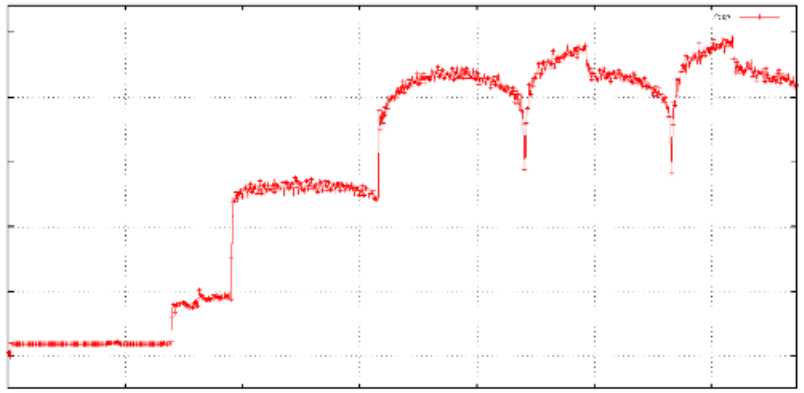
Test patterns

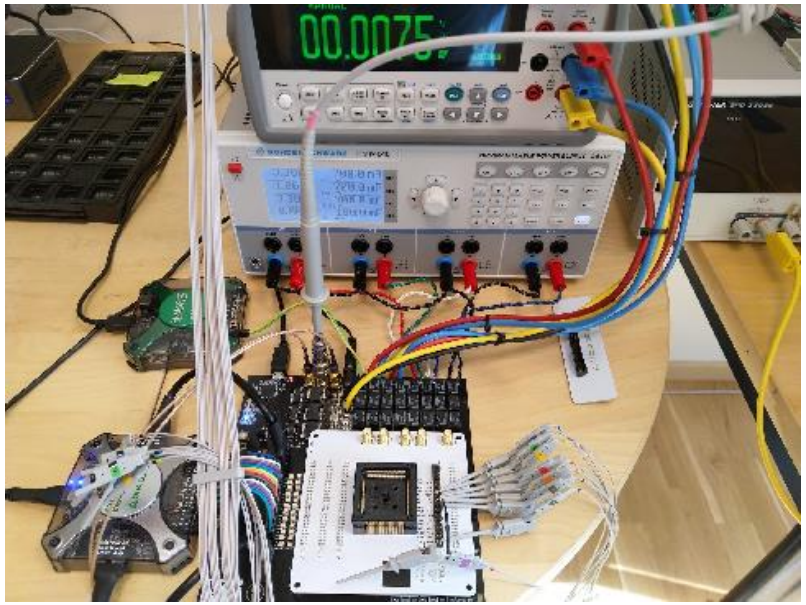
- 70 test patterns
  - 31 acquired from the FPGA prototype
  - 39 synthetically generated patterns



Test method

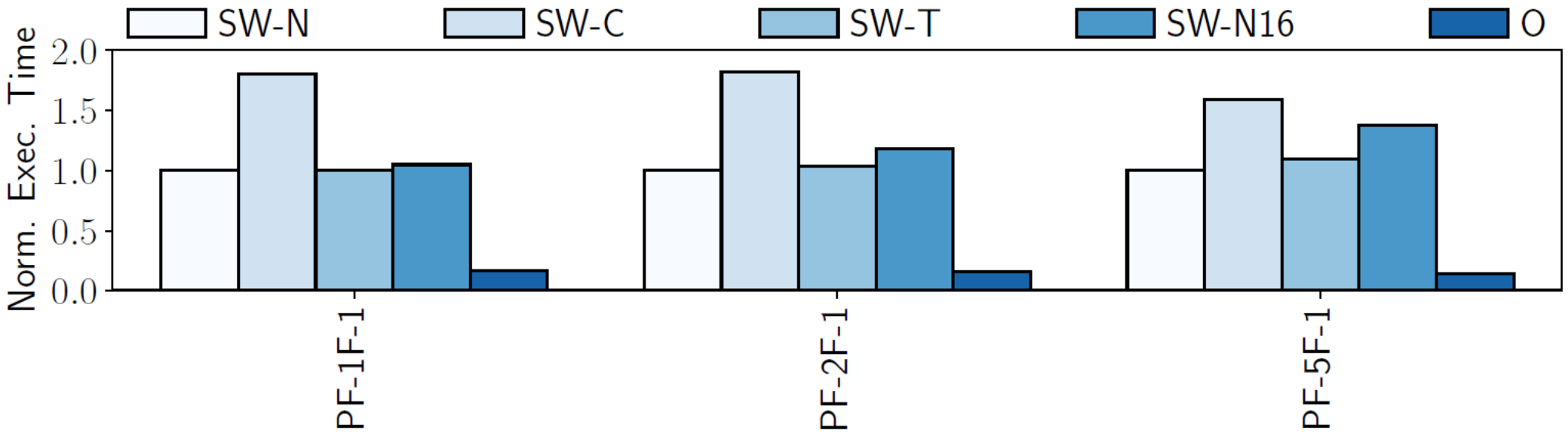
- Test method
  - Simulated on the aXiom chip RTL code
  - RTL current estimation tool
- Measured the actual current consumption on the aXiom prototype
  - Compared against the simulated





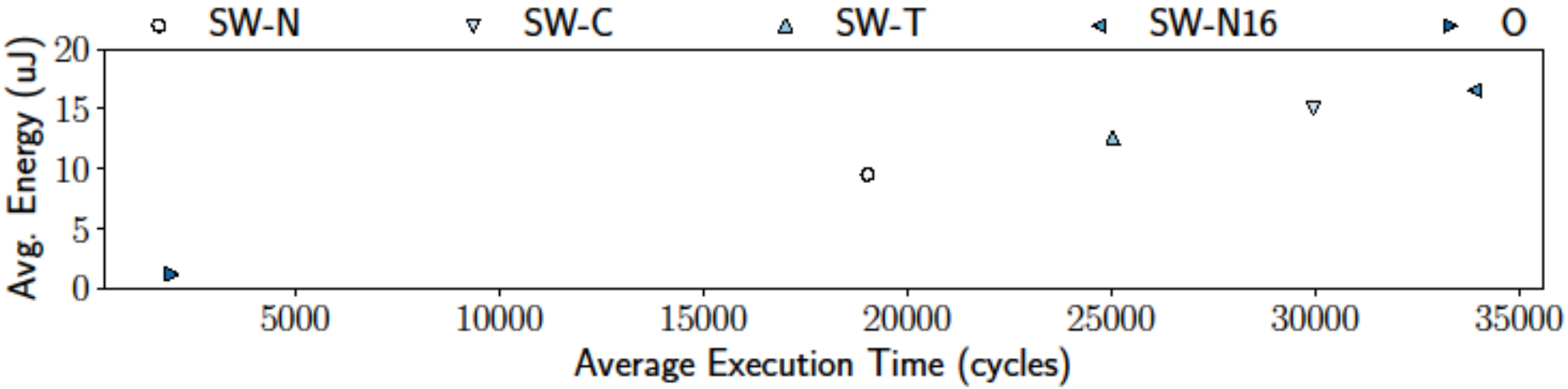
SW results

- The results are data dependent



SW results

- The SW-N algorithm performs best overall
  - The SW-C algorithm is impacted by keeping alive the cache all the time
  - The SW-T and SW-N16 algorithms performs worse when there are nodes over the threshold



- Simulated current consumption correlates with the measured

Outline

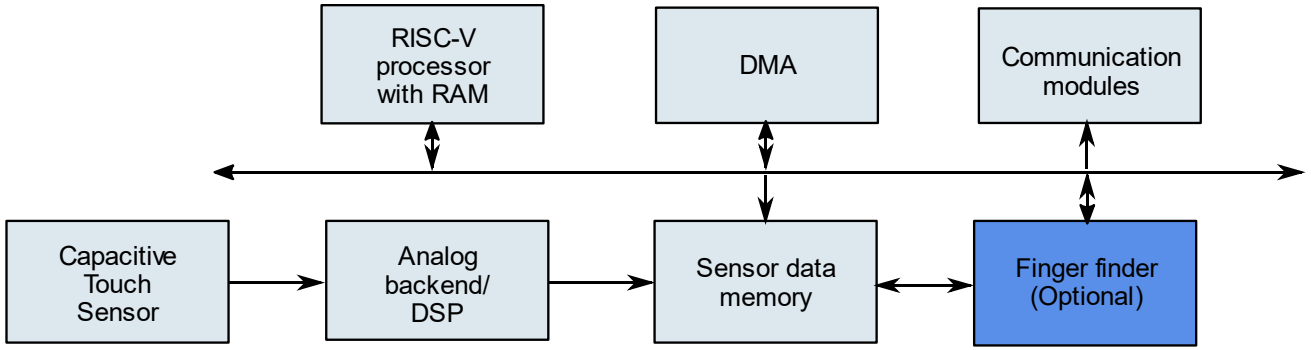
- Introduction
- Peak detection in SW
- **Finger Finder**

Assembly optimizing, RoCC or HW accelerator?

- Looked into assembly optimizing the code

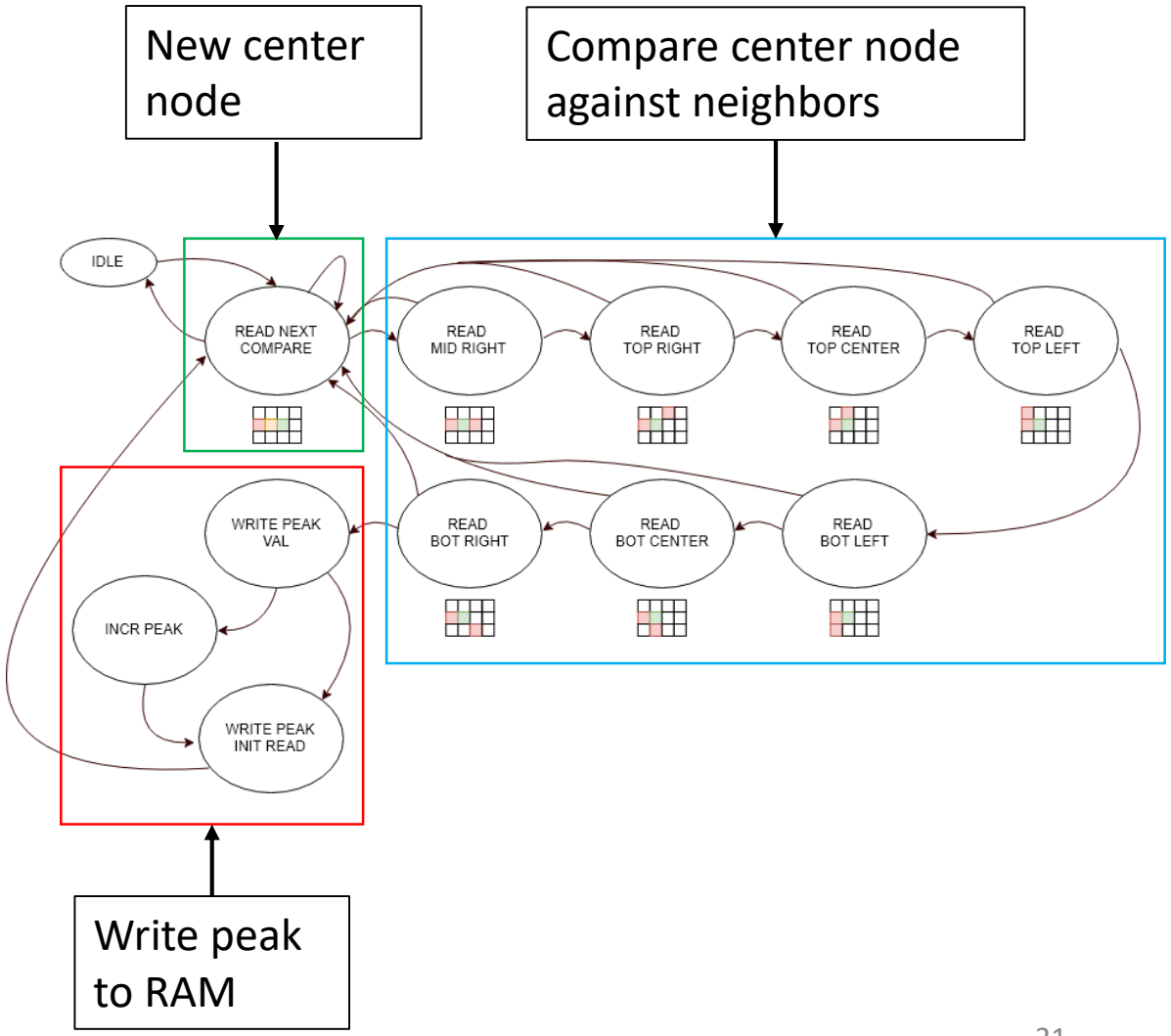
```
400001ea:       43d4                    lw      a3,4(a5)
400001ec:       fea6eae3                bltu    a3,a0,400001e0 <FindPeaks+0x80>
400001f0:       0007a803                lw      a6,0(a5)
400001f4:       ff06e6e3                bltu    a3,a6,400001e0 <FindPeaks+0x80>
400001f8:       0087a303                lw      t1,8(a5)
```

- Looked into making a Rocket Custom Coprocessor via the RoCC interface
- Decided to go for a HW accelerator tied to the sensor data memory



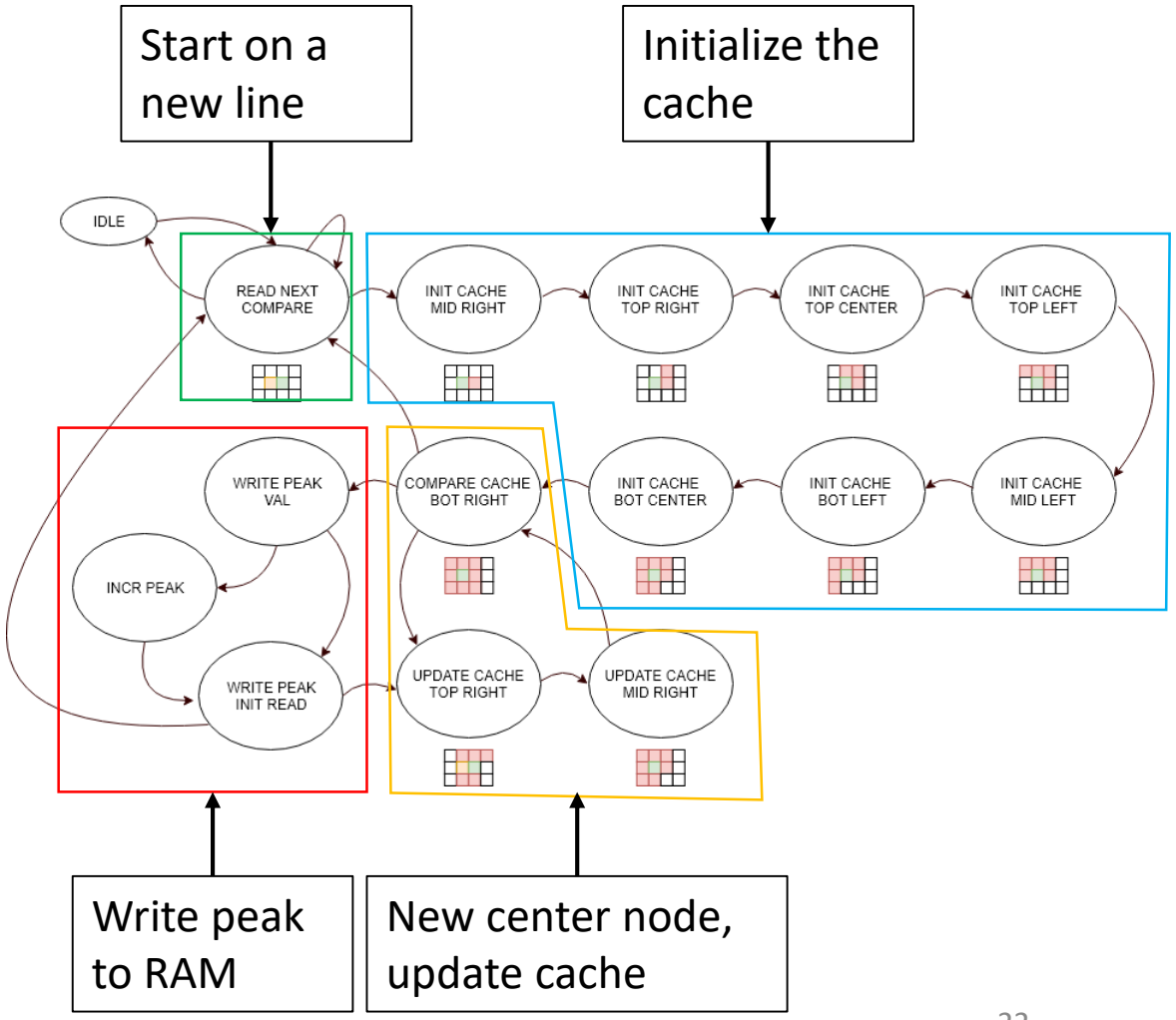
Peak detection algorithms, FF-N

- Implemented the SW-N algorithm as a FSM with direct access to the sensor node data RAM
- When a peak is found the FSM writes the location to a list of peaks in the RAM



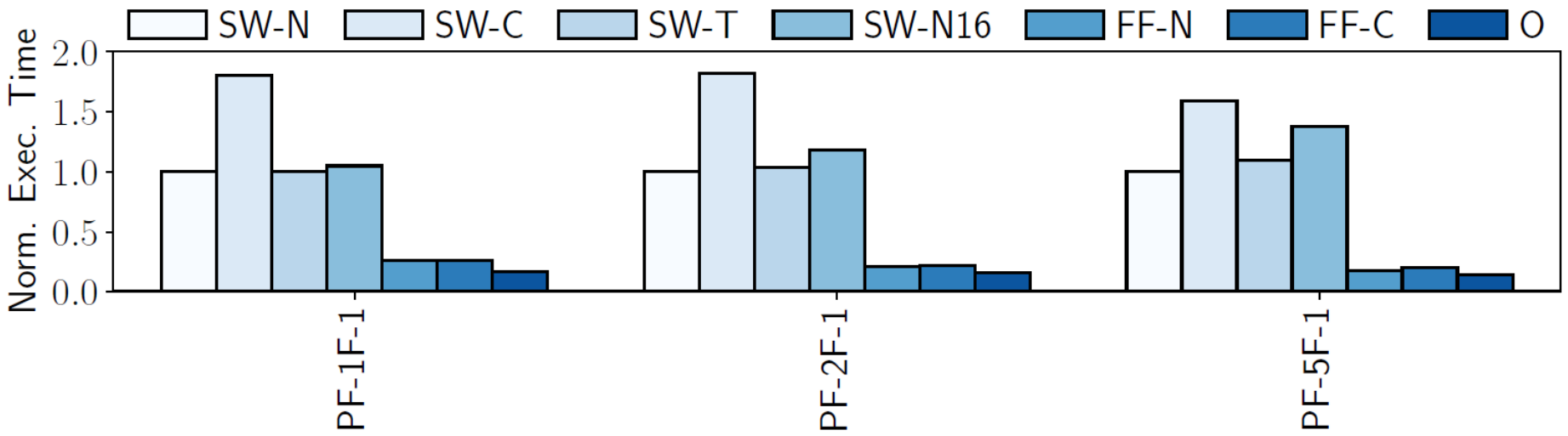
Peak detection algorithms, FF-C

- The FF-C accelerator stores all 8 neighbors in a register bank
  - When going to a new center node the register bank is shifted and three new values are loaded into the bank



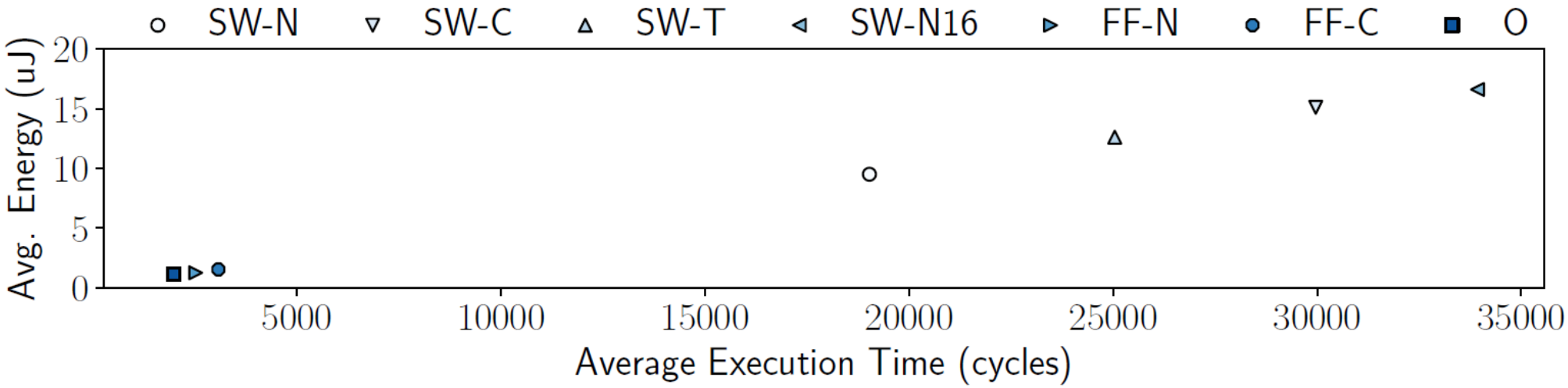
Final results

- Finger finder (FF-N) has the best performance



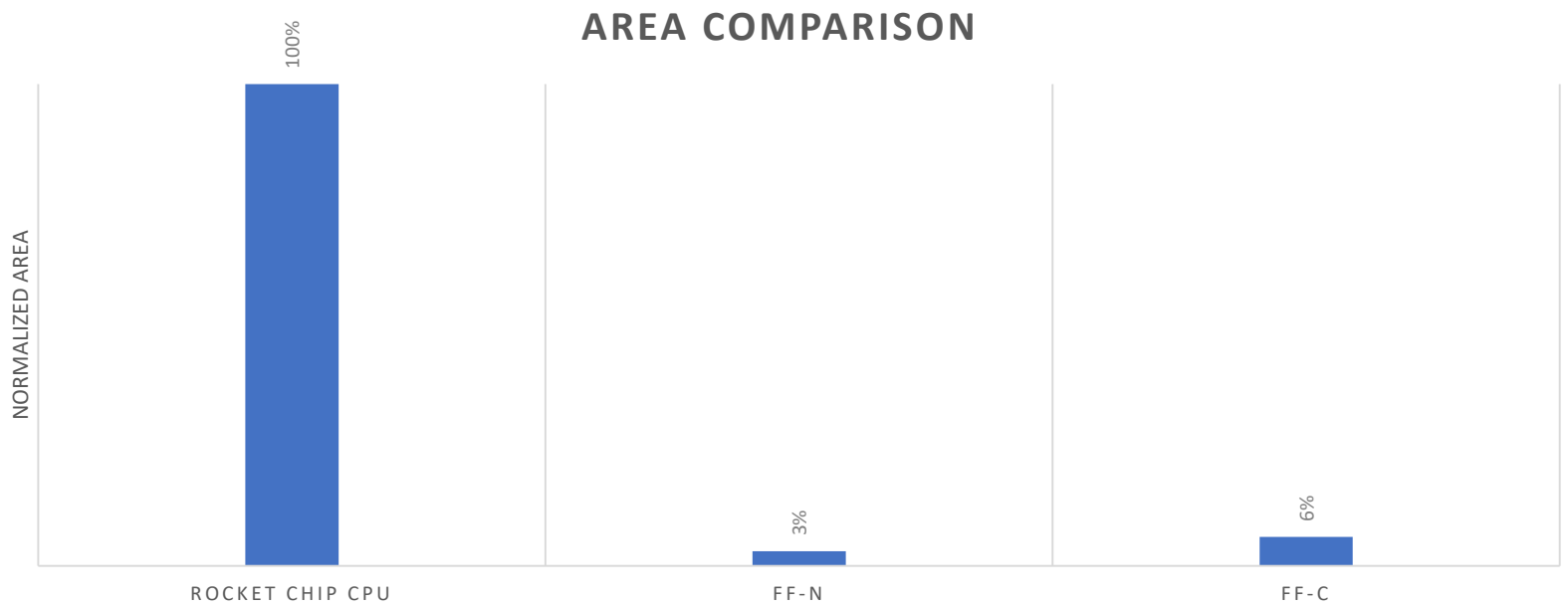
Final results

- Compared SW-N, FF-N reduces the energy consumption with 86%
- 9.5% above the oracle (O)



Final results

- Area impact is small
  - No changes in critical path



Conclusions

- Of the software algorithms, the naïve implementation performs best
- With the Finger Finder HW accelerator we can speed up the peak detection and reduce the power consumption considerable compared to the best SW algorithm

Thank you

Any questions?

Finger Finder: A Low-Energy Peak Detection Accelerator for Capacitive Touch Controllers

- Kai Kristian Amundsen (MyWo AS), Gaute Myklebust (MyWo AS),
- Per Gunnar Kjeldsberg (NTNU) and Magnus Jahre (NTNU)

This work has been supported by RFF Midt-Norge (project 272179).

MyWo AS

REGIONALE FORSKNINGSFOND MIDT-NORGE

NTNU
Norwegian University of Science and Technology