SINE WAVE GENERATOR

Md. Aamir Sohail, Piyush Udhan

EE5811: FPGA LAB
&
EE3025: Independent Project

Department of Electrical Engineering,
IIT HYDERABAD

April 25, 2019

Project Outline

The aim of our project is to make a sine wave generator using Icoboard FPGA. We have used cordic algorithm to do so along with arduino Mega for post processing. To realise the sine wave, we have used an R2R DAC circuit to convert digital signal to analog and a digital oscilloscope (Analog Discovery) to see the waveform on the screen.

- All the verilog and arduino codes, the report and presentation are available on the github link which is a free open-source. ▸ Github Link

Software setup

1. Open the command terminal execute the following command ▸ Link for installing: [1]
   - wiringPi
   - IcoProg
   - IcoStorm tools
   - Arachne-pnr
   - Yosys
2. Creating **Makefile** [1]

MOTIVATION

If a unit vector with co-ordinates $(x_1, y_1) = (1, 0)$ is rotated by an angle θ, its new co-ordinate will be $(x_2, y_2) = (\cos\theta, \sin\theta)$. Thus, by finding the (x_2, y_2), $\cos\theta, \sin\theta$ can easily be computed.

Circular CORDIC Algorithm

CORDIC iteration: [2]

$$x(i+1) = x(i) - d_i\, y(i)\, 2^{-i} \tag{1}$$

$$y(i+1) = y(i) + d_i\, x(i)\, 2^{-i} \tag{2}$$

$$\alpha(i+1) = \alpha(i) - d_i\, \tan^{-1} 2^{-i} \tag{3}$$

Choose $d_i \in \{+1, -1\}$ such that $\alpha(n) \to 0$

After n pseudorotation steps, when $\alpha(n)$ is well enough close to zero, we will get $\sum \theta(i) = \alpha$. Finally the CORDIC iterations in ROTATION MODE become:

$$x(n) = \mathbf{K}\,(x \cos\alpha - y \sin\alpha) \tag{4}$$

$$y(n) = \mathbf{K}\,(y \cos\alpha + x \sin\alpha) \tag{5}$$

$$\alpha(n) = 0 \tag{6}$$

where $\mathbf{K} = \prod(1 + \tan^2\theta(i))^{1/2}$, $\theta(i) = \tan^{-1}(d_i\, 2^{-i})$

Domain of Convergence

- In rotation mode, convergence of $\alpha(n)$ to 0 is possible because each CORDIC angle is more than half the previous angle.

$$\tan^{-1}(2^{-(i+1)}) \geq 0.5 * \tan^{-1}(2^{-i})$$

- Thus, domain of convergence is $-99.7^o \leq \alpha \leq 99.7^o$ where $\pm 99.7^o$ is the sum of the CORDIC angle.
- Beyond the above range, we can use Trigonometric identities to convert the problem within the domain of convergence.

Implementation of Circular CORDIC in RTL

- The digital design should be able to compute Sine and Cosine of the input angles $\in [0, 2\pi]$ including the floating point angles like 89.45^o, 29.7^o etc.
- A 16-bit binary scaling system has been used to represent angles. The resolution of the design is $360^o/2^{16} = 5.49316406 \text{ X } 10^{-3}$. Angle^o to 16-bit binary value conversion:

$$(16bit)\mathbf{angle} = \frac{2^{16} * angle^o}{360^o}$$

- The upper two bits represent the quadrant [3].
  - 2b'00 -> represents I quadrant i.e (0 - $\pi/2$) range
  - 2b'01 -> represents II quadrant i.e ($\pi/2$ - π) range
  - 2b'10 -> represents III quadrant i.e (π - $3\pi/2$) range
  - 2b'11 -> represents IV quadrant i.e ($3\pi/2$ - 2π) range

Coding and Hardware Environment

- CORDIC Algorithm: Verilog
  1. Xilinx ISE-Design Suite 14.7[1]
  2. Simulation: Xilinx-ISim software

- Postprocessing:
  1. Software: Arduino ▸ Download Link
  2. Board: Arduino MEGA 2560

- Oscilloscope:
  1. DAC: 'R2R Ladder' ▸ Link
  2. Analog Discovery (Digital Oscilloscope) [2]
     - Software: Waveform ▸ Download Link

[1] L.Gregg - How to Download & Install Xilinx ISE 14.7 Windows 10 ▸ Youtube Link

[2] Getting Started with Analog Discovery ▸ Youtube Link

Circuit and System Setup

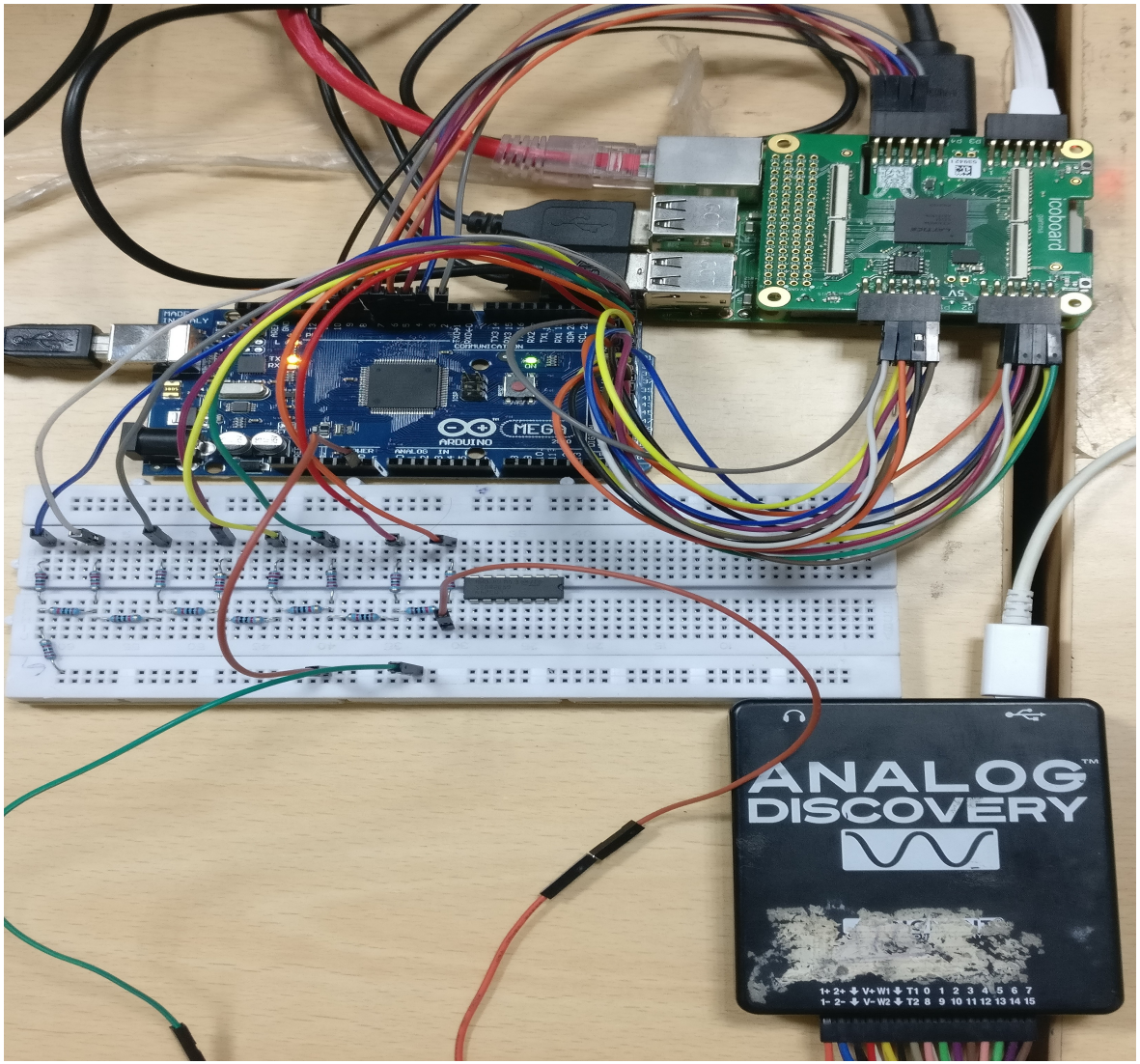

Figure: Circuit Arrangement

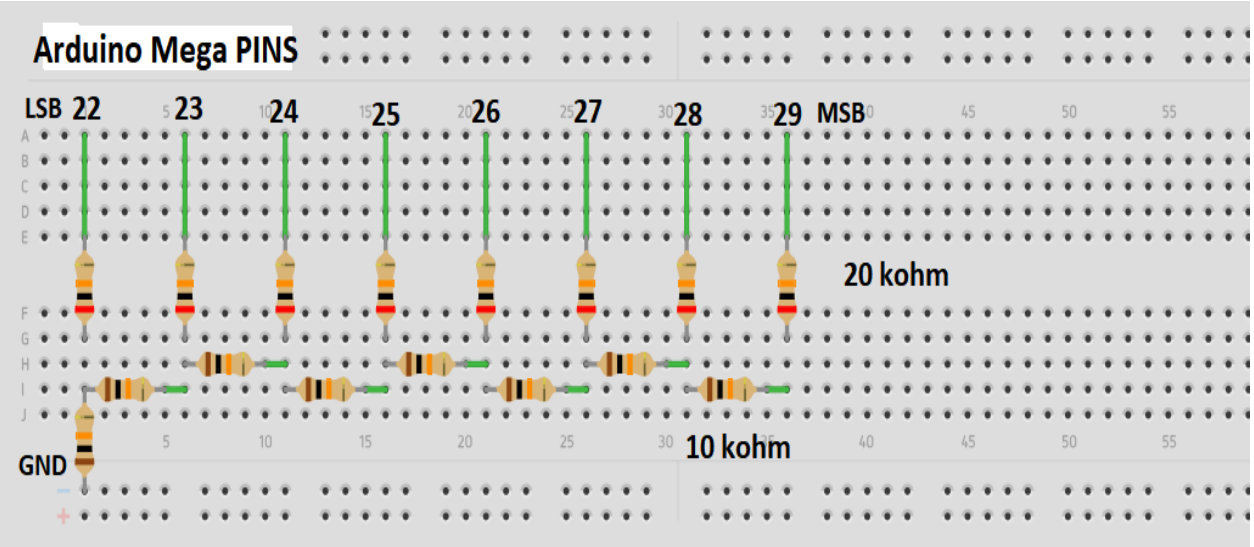

Figure: 8-bit 'R2R' DAC Circuit

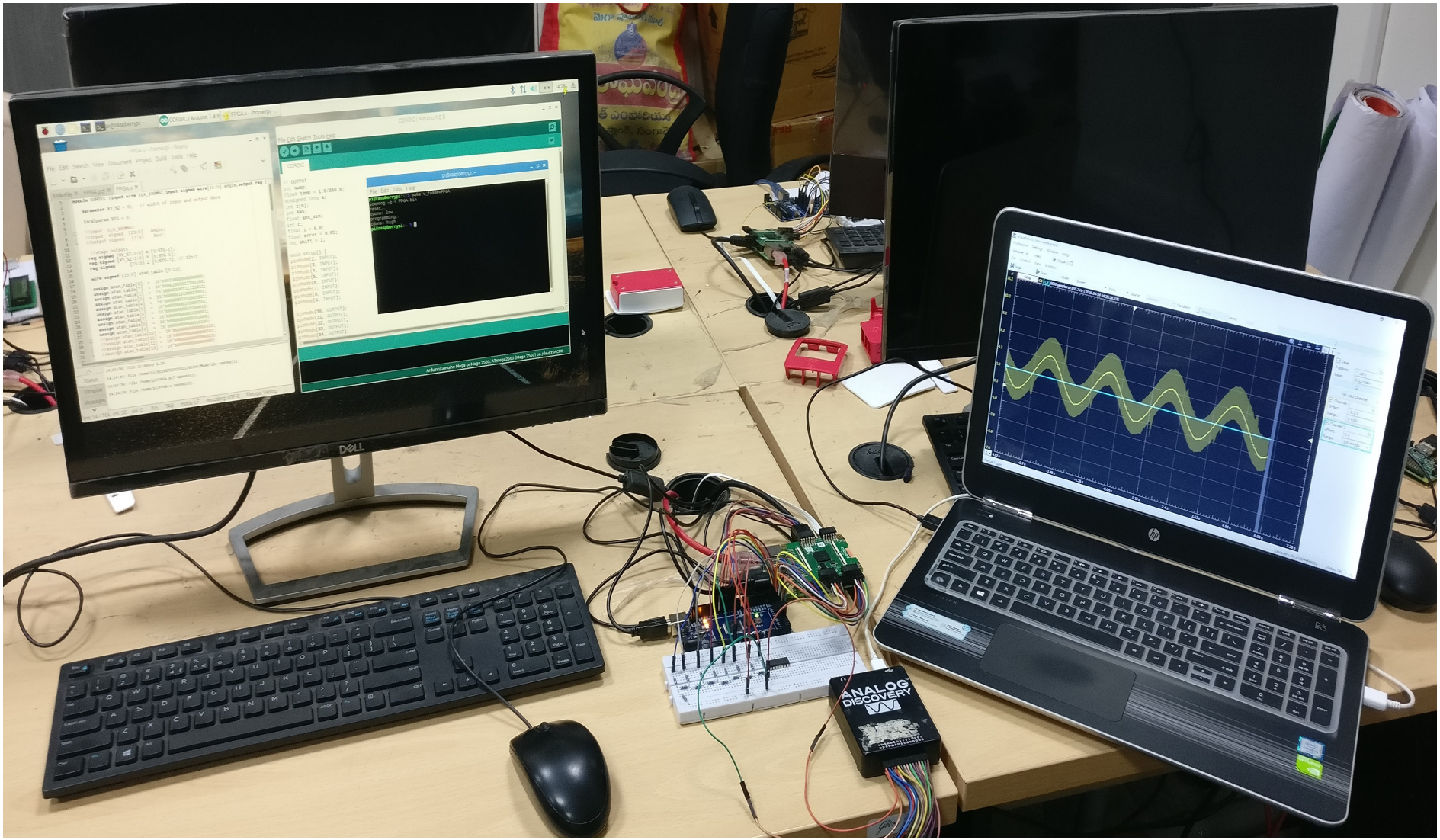

Figure: System Setup

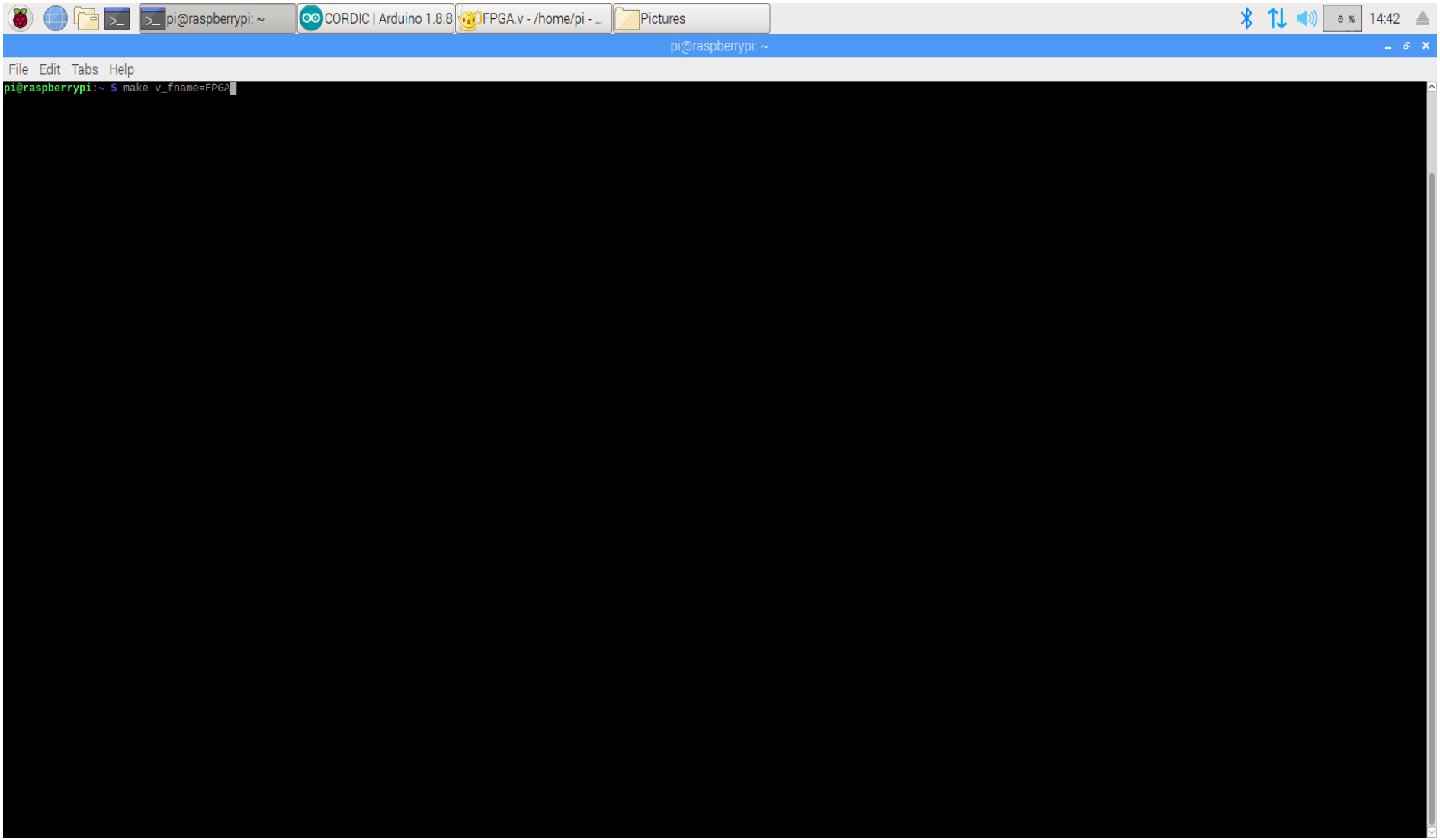

Figure: Uploading FPGA.v to Ico-Board FPGA

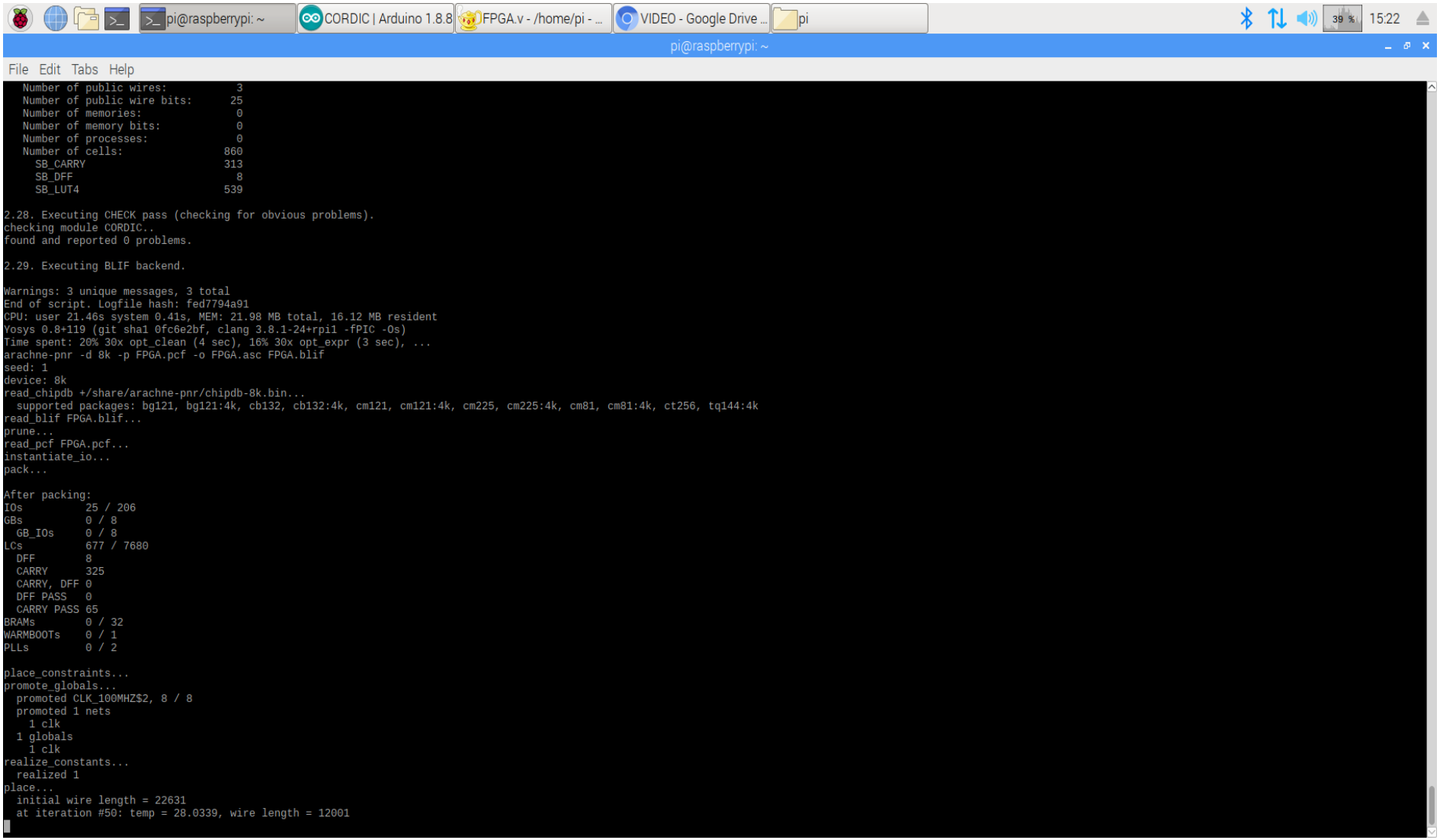

Figure: Uploading FPGA.v to Ico-Board FPGA

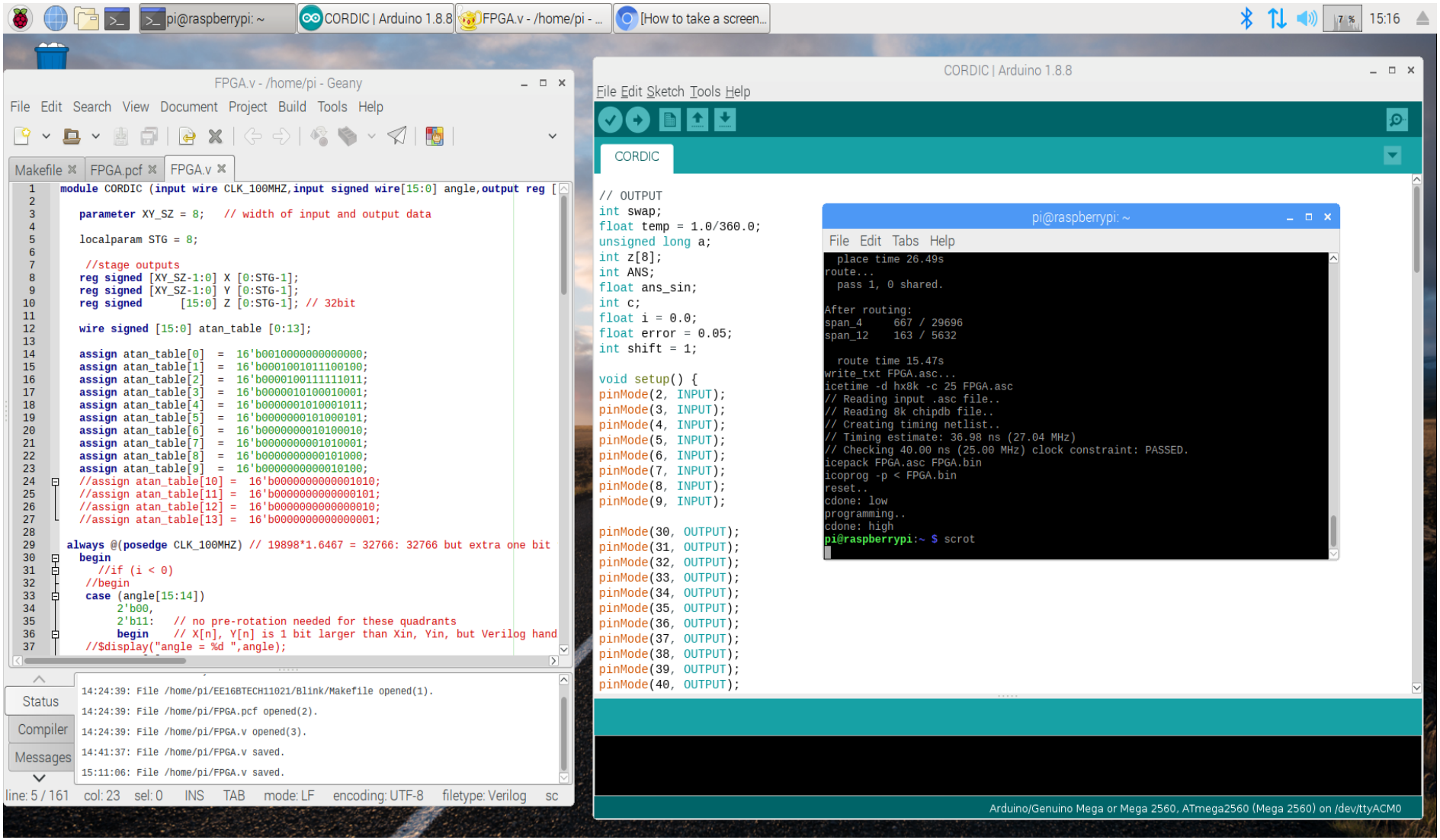

Figure: Uploading FPGA.v to Ico-Board FPGA

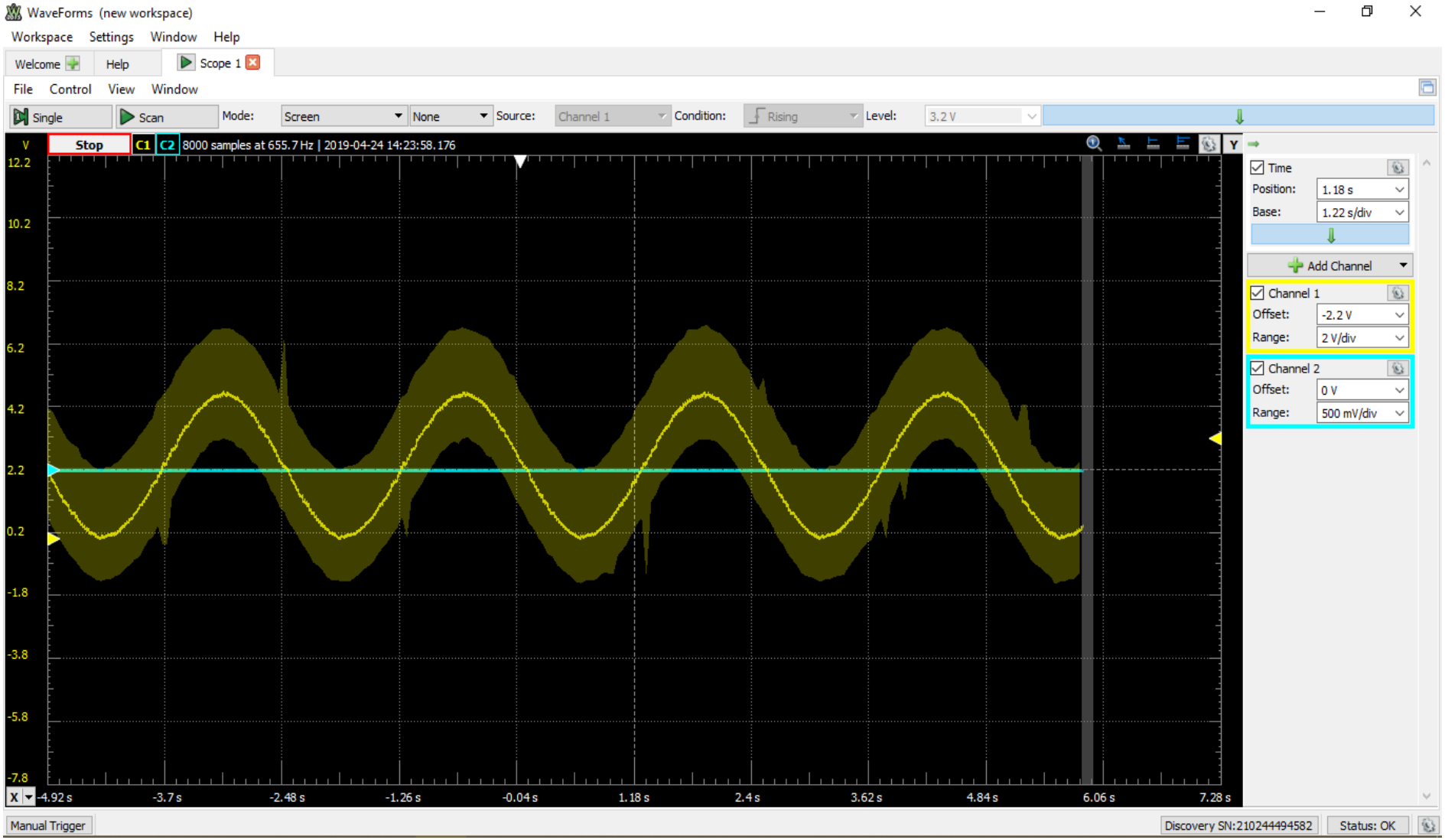

Figure: Output Waveform

References


GVV Sharma, Hemanth Kumar Desinnedi,
"Introduction to verilog programming on IcoBoard FPGA", ▸ Link


Behrooz.Parhami., COMPUTER ARITHMETIC-Algorithms and Hardware Designs 2nd edition., OXFORD UNIVERSITY PRESS, 2010, ch.22.


Kirk Weedman, CORDIC Design and Simulation using Verilog, ▸ Link