CHALMERS
UNIVERSITY OF TECHNOLOGY

2020-01-30 Technical information

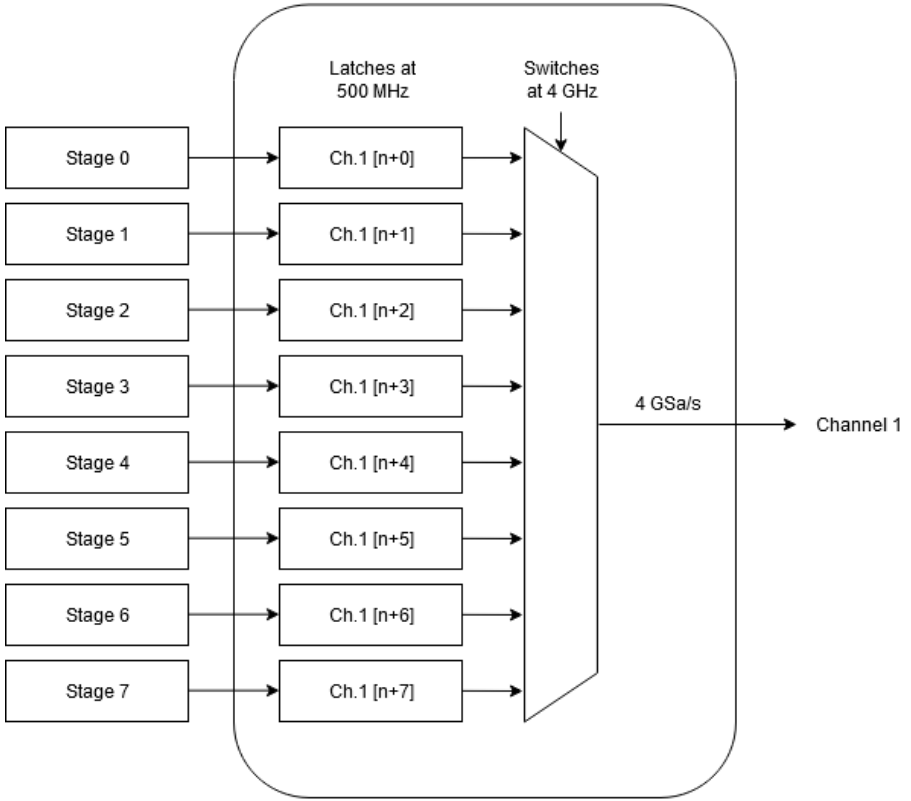
- **Upcoming specs, clarifications**
- **Ethernet interfacing**
- **Questions and their answers**

Christian Križan
Research engineer, QTL
krizan@chalmers.se

Specification documents incoming

- Hints on testing
- Tips for concepts to research
- Prioritisation
- Articles for various modules, sinusoidal generation
- (Maybe?) Tips and ideas for handling dropped packages

Clarification on the DAC



Ethernet interfacing

- The Nexys 4 board carries a PHY chip (LAN8720). Its pin routing to the Artix 7 FPGA is given in the Nexys 4 manual, interfacing details are found in a LAN8720 datasheet.

  For this project, we have specified that you interface to this PHY via RMII.

- The resulting bitstream will (hopefully) contain Ethernet frames.
  In such a frame there will be a bitlength designated (‘by standards’) as an IP packet.

- Within that packet, you will find a UDP packet.

- The header of this packet contains the destination port, which you will use to figure out which target DAC FIFO this packet’s payload is destined for. The ensuing data (and its format) is given in the module descriptive document in Canvas.

Cont. payloads

- Everything else that does not conform to the expected stream data, send it to other_data_tx. This includes control words.

- These control words will have to be parsed by a control word host. Its output is expected to be routed to various blocks in your overall architecture.
  Question: Would you like a module graphic for this host?

- OK, but what about ARP?
  The ARP request is contained within the IP Datagram field in the previous slide.

Address Resolution Protocol

- Every node on the network has its own MAC address. But applications communicate using IP addresses. Thus at some level, we must link MAC addresses to IP addresses.

- The ARP request is basically shouting on the network
  'Whom has the IP address 003.133.133.13' ?

  (… and whom you're supposed to yell the answer to.)

- The guilty node responds to the specific address. All other nodes stay quiet.
  In this project, you'll handle this ARP tango in a separate block within the Ethernet communications module.

- Manual MAC to (static) IP linking can be done manually in your OS of choice, but it's a delicate matter.

  Which is the whole point of including this block in this project.

Suggested implementation steps

1. Begin by interfacing to the Nexys 4 PHY
2. Dissect a faked payload and fetch its status (UDP or not?)
3. Add discriminator which sends everything in port range 30000 – 30255 onto some data bus
4. Add ARP request block, signal whether an ARP package was detected
5. Try to do everything above in reverse, loopback the data bus, and monitor on the PC using some serial monitoring program (like Wireshark)
6. Add missing functionality, although without AXI
7. Get AXI working, the bus you'll use for all other intra-'big module'-communication

Answers to questions

Answers to questions

- Stream ID bus
- The RLE encoding
- Packet sequence numbering
- How to target a DAC channel
- Ethernet data structure
- Windowing
- Sum block
- Research papers for the modules (NCO!)
- Time needed in this project for every module

“The minimum system”

- Ultra-bare minimum that can run on the QPU:
  - 1. Ethernet module except ARP functionality.
  - 2. Naïve, dumb and slow stream management without DDR. Basically EDF constrained (earliest-deadline-first)
  - 3. Two upconversion stages with: FIFO, no RLE (just loose samples…), no IQ mixing ergo no NCO, **BUT a single target upconversion set in the DAC itself,** no summation (has to be done by the PC before-hand (slow)).
  - 4. A single downconversion stage with: no IQ mixing ergo no NCO, **BUT a single target downconversion set in the ADC itself,** no skip/store, but yet a FIFO at the end.

- What can you do with that? Actually quite a lot. More than this was not needed very recently to verify that the qubits you’ll be using are working.

- Also, completion of any one of the four major modules yields something that is usable.

QPU control signalling

Multi-Qubit Pulse Generator - MME

Trace: All | Get trace | Save trace | Start | Stop

Sections
- Communication
- Sequence
- Waveform
- 1-QB gates XY
- 1-QB gates Z
- QB spectra
- 2-QB gates
- Tomography
- Predistortion
- Cross-talk
- Readout
- Output
- Demodulation

Readout

Readout pulse type: Cosine
- [x] Uniform amplitude
- Amplitude [V]: 100E-3
- [x] Uniform pulse shape
- Width [s]: 10E-9
- Duration [s]: 20E-9
- [x] Match main sequence waveform size
- [ ] Distribute readout phases
- Readout delay [s]: 0
- Readout I/Q ratio: 1
- Offset I [V]: 0
- Offset Q [V]: 0
- IQ skew [deg]: 0

Readout #1
- Frequency [Hz]: 100E6

Readout #2
- Frequency [Hz]: 0

Readout #3
- Frequency [Hz]: 0

Readout #4
- Frequency [Hz]: 0

Readout #5
- Frequency [Hz]: 0

Readout #6
- Frequency [Hz]: 0

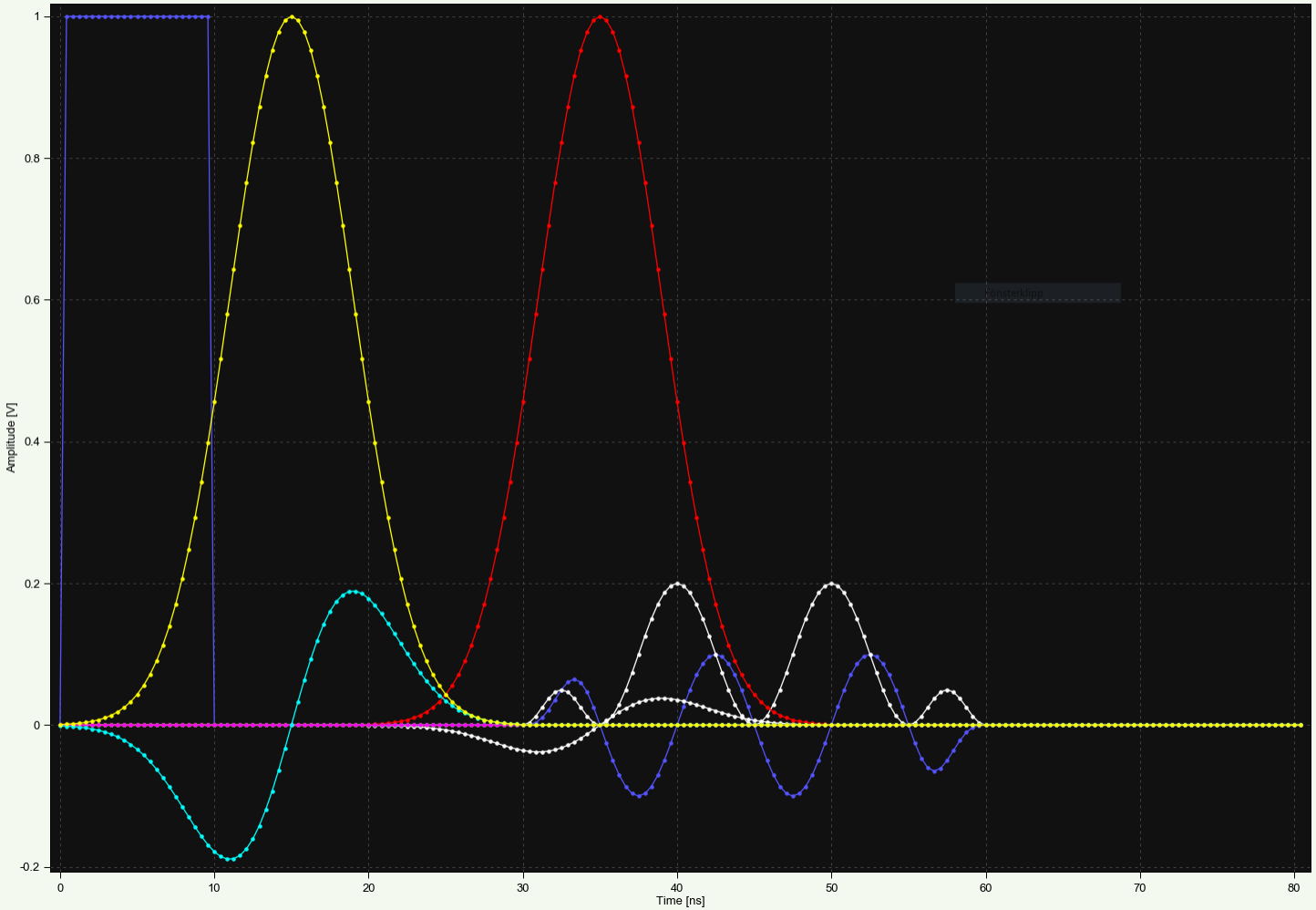
Readout #7
- Frequency [Hz]: 0

Readout #8
- Frequency [Hz]: 0

Readout #9
- Frequency [Hz]: 0

Readout predistortion
- [ ] Predistort readout waveform

Readout trig
- [ ] Generate readout trig
- Readout trig amplitude [V]: 1
- Readout trig duration [s]: 20E-9



Send/receive | Active

Multi-Qubit Pulse Generator - MME

Trace: Trace - I1 | Get trace | Save trace | Start | Stop

Sections
- Communication
- Sequence
- Waveform
- 1-QB gates XY
- 1-QB gates Z
- QB spectra
- 2-QB gates
- Tomography
- Predistortion
- Cross-talk
- Readout
- Output
- Demodulation

Readout

Readout pulse type: Cosine
- [x] Uniform amplitude
- Amplitude [V]: 100E-3
- [x] Uniform pulse shape
- Width [s]: 10E-9
- Duration [s]: 20E-9
- [x] Match main sequence waveform size
- [ ] Distribute readout phases
- Readout delay [s]: 0
- Readout I/Q ratio: 1
- Offset I [V]: 0
- Offset Q [V]: 0
- IQ skew [deg]: 0

Readout #1
- Frequency [Hz]: 100E6

Readout #2
- Frequency [Hz]: 0

Readout #3
- Frequency [Hz]: 0

Readout #4
- Frequency [Hz]: 0

Readout #5
- Frequency [Hz]: 0

Readout #6
- Frequency [Hz]: 0

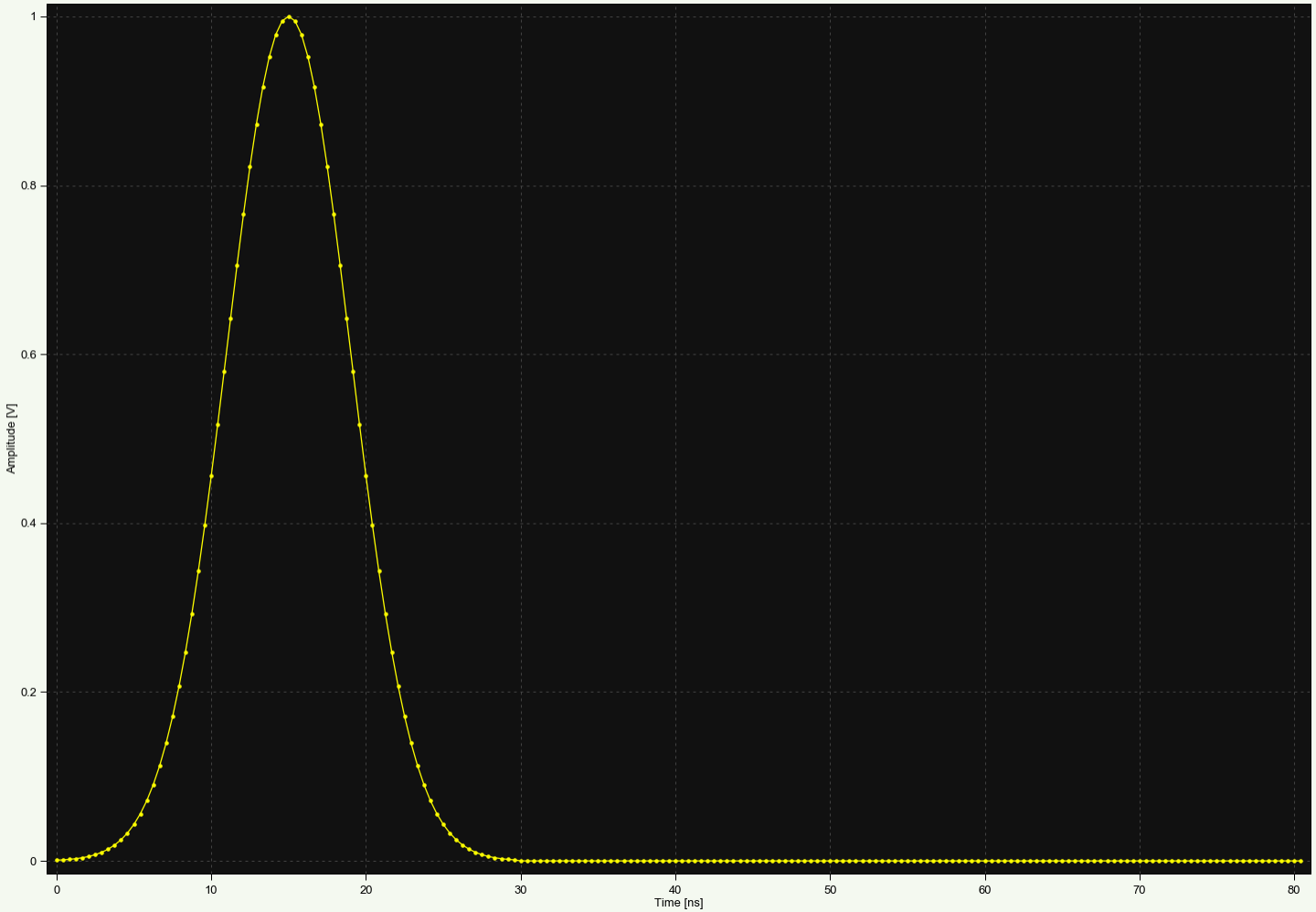
Readout #7
- Frequency [Hz]: 0

Readout #8
- Frequency [Hz]: 0

Readout #9
- Frequency [Hz]: 0

Readout predistortion
- [ ] Predistort readout waveform

Readout trig
- [ ] Generate readout trig
- Readout trig amplitude [V]: 1
- Readout trig duration [s]: 20E-9



Send/receive | Active

QPU readout

Multi-Qubit Pulse Generator - MME

Trace: Trace - Readout I | Get trace | Save trace | Start | Stop

Sections
- Communication
- Sequence
- Waveform
- 1-QB gates XY
- 1-QB gates Z
- QB spectra
- 2-QB gates
- Tomography
- Predistortion
- Cross-talk
- Readout
- Output
- Demodulation

Readout
- Readout pulse type: Square
- Uniform amplitude
- Amplitude [V]: 100E-3
- Uniform pulse shape
- Width [s]: 1E-6
- Duration [s]: 20E-9
- Match main sequence waveform size
- Distribute readout phases
- Readout delay [s]: 0
- Readout I/Q ratio: 1
- Offset I [V]: 0
- Offset Q [V]: 0
- IQ skew [deg]: 0

Readout #1
- Frequency [Hz]: 100E6

Readout #2
- Frequency [Hz]: 0

Readout #3
- Frequency [Hz]: 0

Readout #4
- Frequency [Hz]: 0

Readout #5
- Frequency [Hz]: 0

Readout #6
- Frequency [Hz]: 0

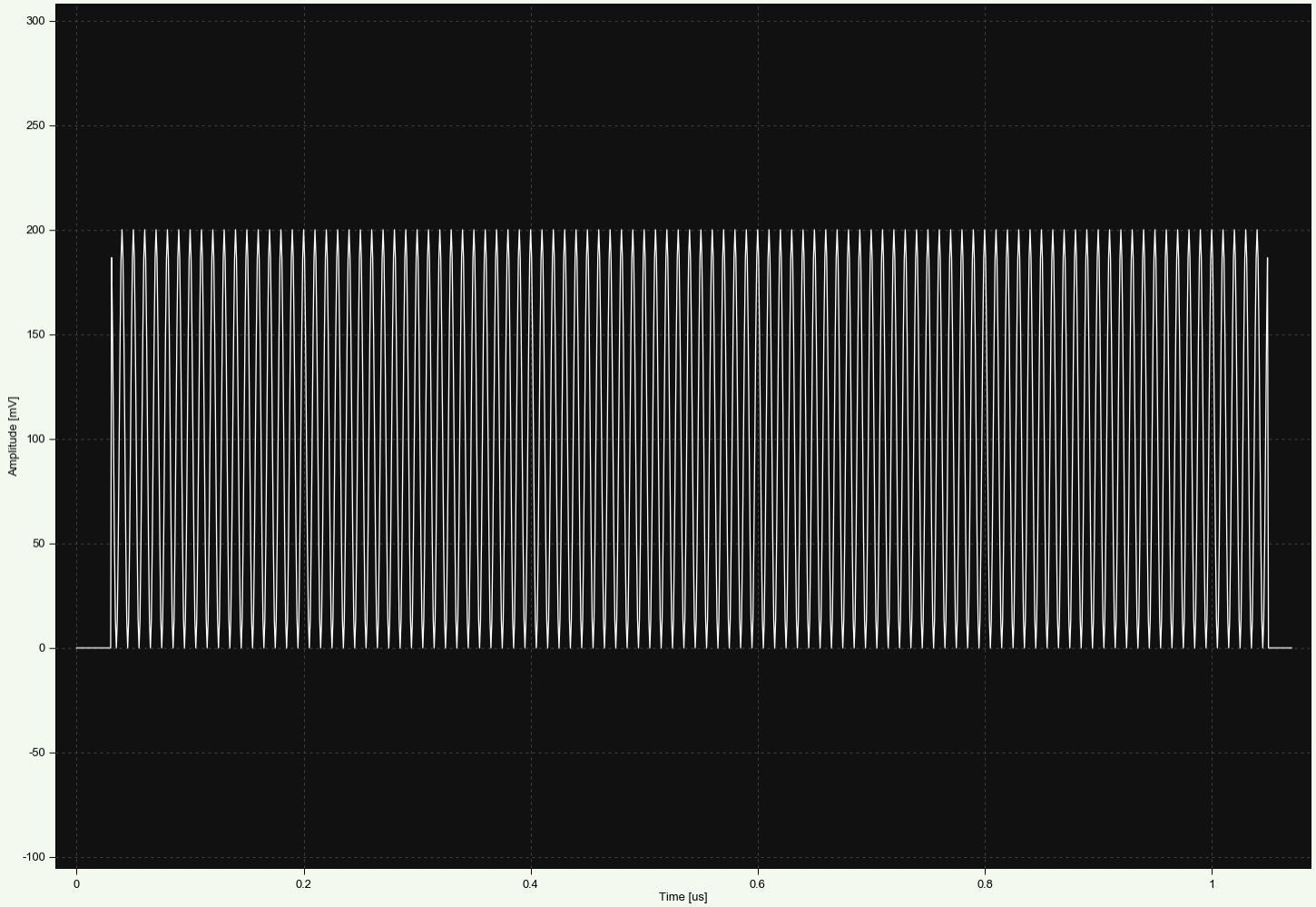
Readout #7
- Frequency [Hz]: 0

Readout #8
- Frequency [Hz]: 0

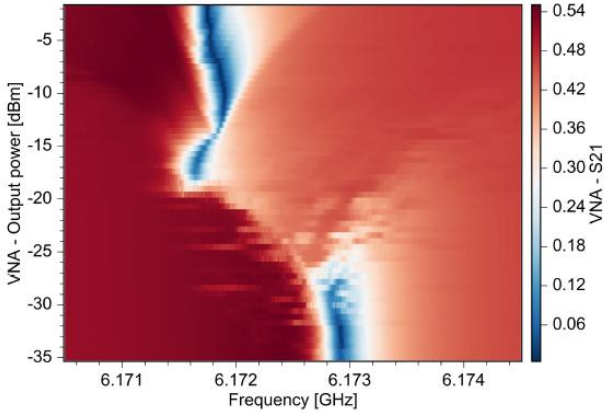
Readout #9
- Frequency [Hz]: 0

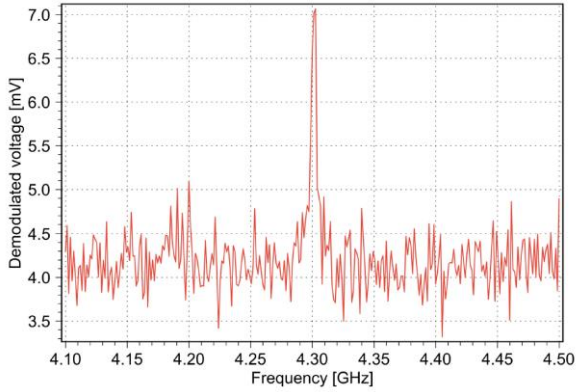
Readout predistortion
- Predistort readout waveform

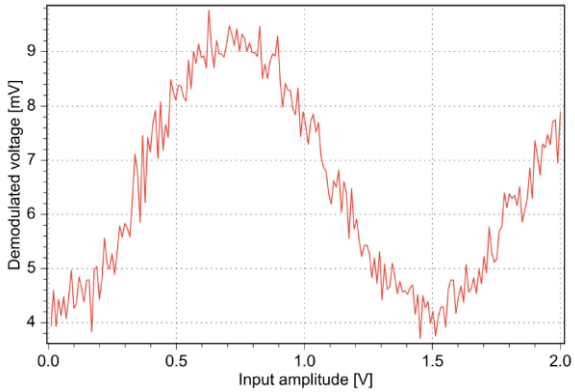
Readout trig
- Generate readout trig
- Readout trig amplitude [V]: 1
- Readout trig duration [s]: 20E-9

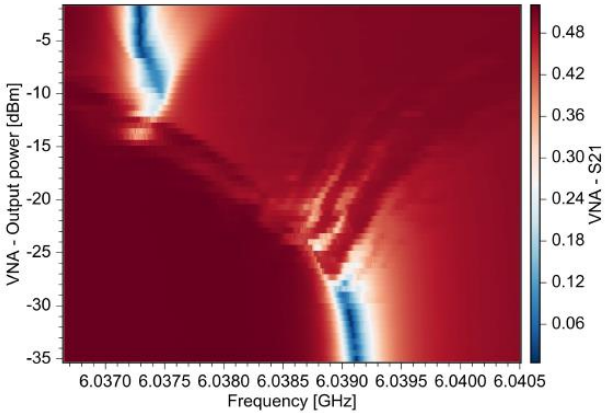


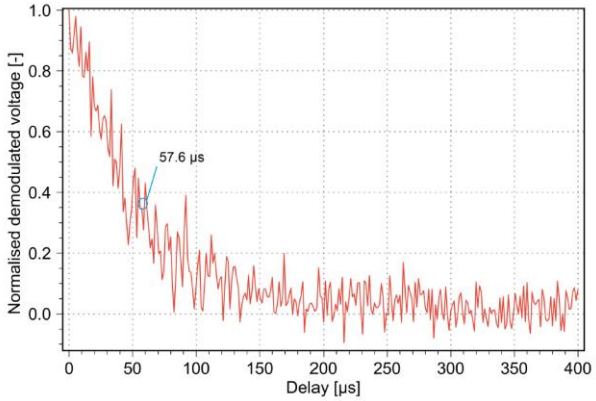
Send/receive | Active

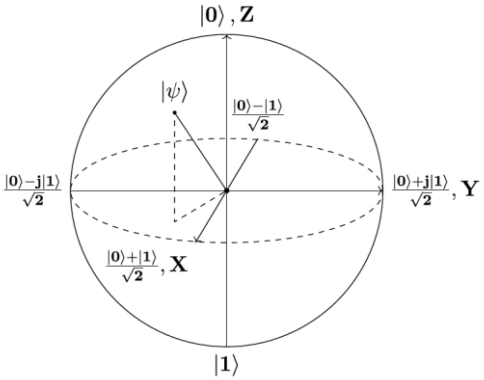

VNA - Output power [dBm]
-5
-10
-15
-20
-25
-30
-35
6.171
6.172
6.173
6.174
Frequency [GHz]
VNA - S21
0.54
0.48
0.42
0.36
0.30
0.24
0.18
0.12
0.06


Demodulated voltage [mV]
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
4.10
4.15
4.20
4.25
4.30
4.35
4.40
4.45
4.50
Frequency [GHz]


Demodulated voltage [mV]
9
8
7
6
5
4
0.0
0.5
1.0
1.5
2.0
Input amplitude [V]

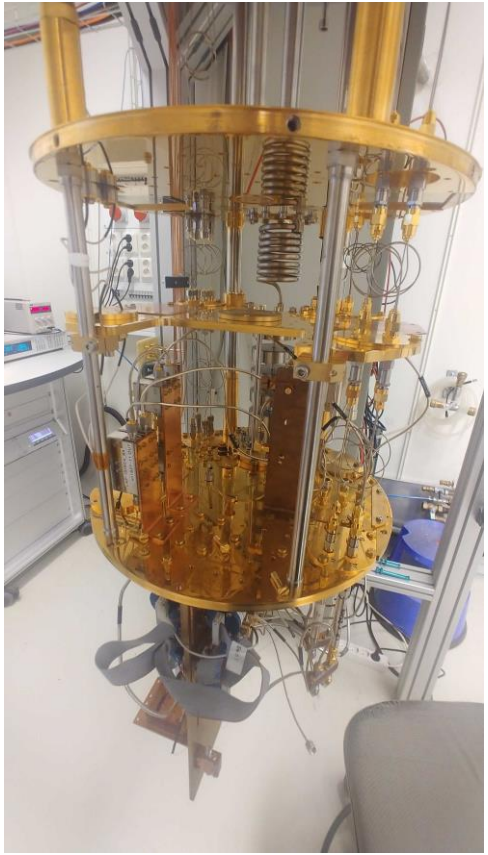

VNA - Output power [dBm]
-5
-10
-15
-20
-25
-30
-35
6.0370
6.0375
6.0380
6.0385
6.0390
6.0395
6.0400
6.0405
Frequency [GHz]
VNA - S21
0.48
0.42
0.36
0.30
0.24
0.18
0.12
0.06

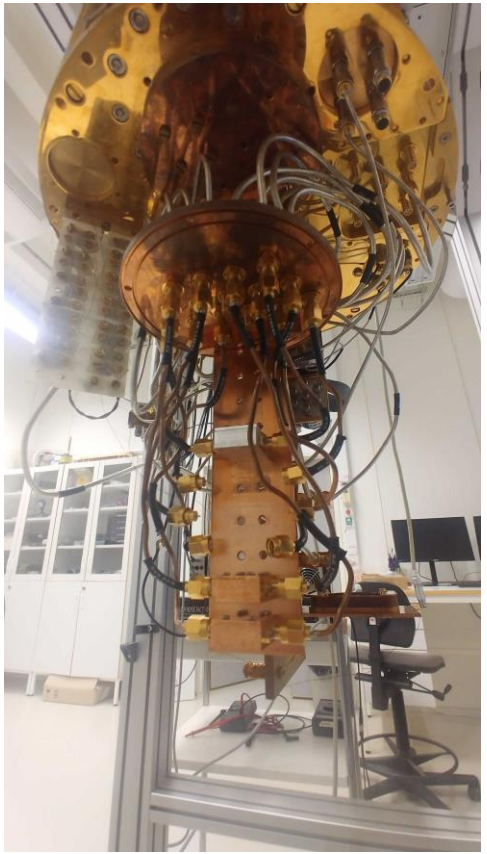

Normalised demodulated voltage [-]
1.0
0.8
0.6
0.4
0.2
0.0
57.6 µs
0
50
100
150
200
250
300
350
400
Delay [µs]


|0⟩ , Z
|ψ⟩
|0⟩−|1⟩ / √2
|0⟩−j|1⟩ / √2
|0⟩+j|1⟩ / √2 , Y
|0⟩+|1⟩ / √2 , X
|1⟩

Visit?

Working on it





Contract?

Open-class questions

Or swoosh me a pigeon to krizan@chalmers.se or something :)

CHALMERS

UNIVERSITY OF TECHNOLOGY