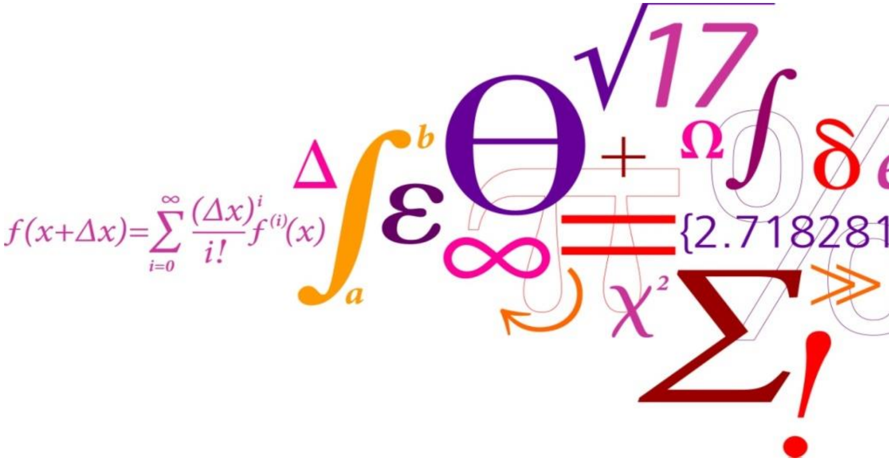
Research scanners at CFU

by Borislav Tomov,
CFU



Center for Fast Ultrasound Imaging
Department of Health Technology

Why

- Commercial scanner : video out
- Comm. scanner + research interface : RF BF data
- Research scanner: full access to setup and RF channel data

How

- Setup of emission sequences (frames)
- Setup of transmit
- Setup of receive
- Setup of image processing/navigation

2-channel sampling system (1991)

ADC:
- 20MHz
- 12-bit

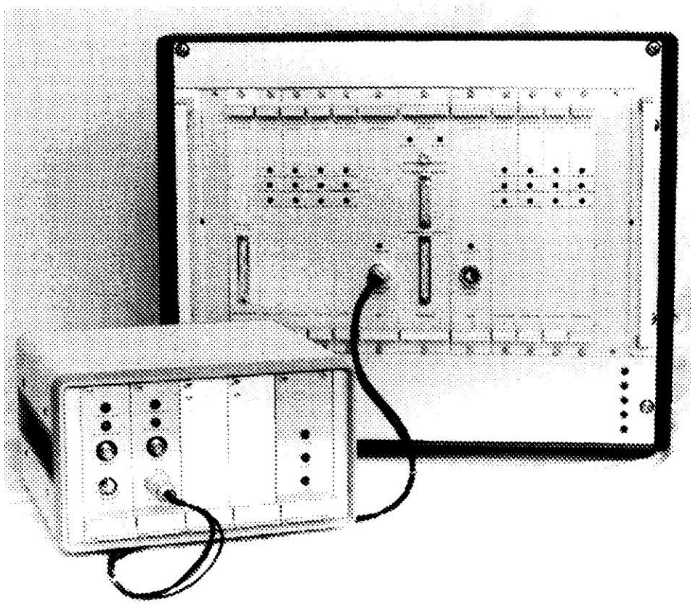
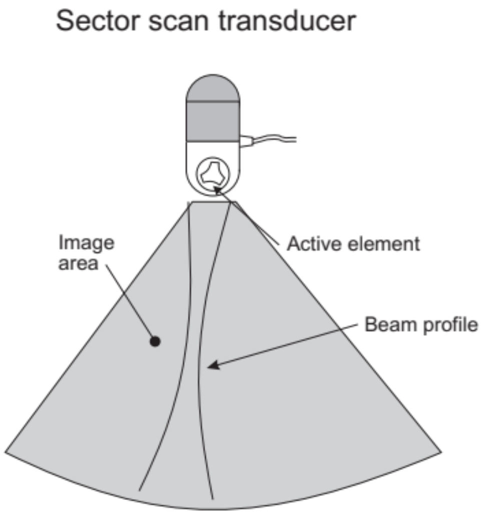


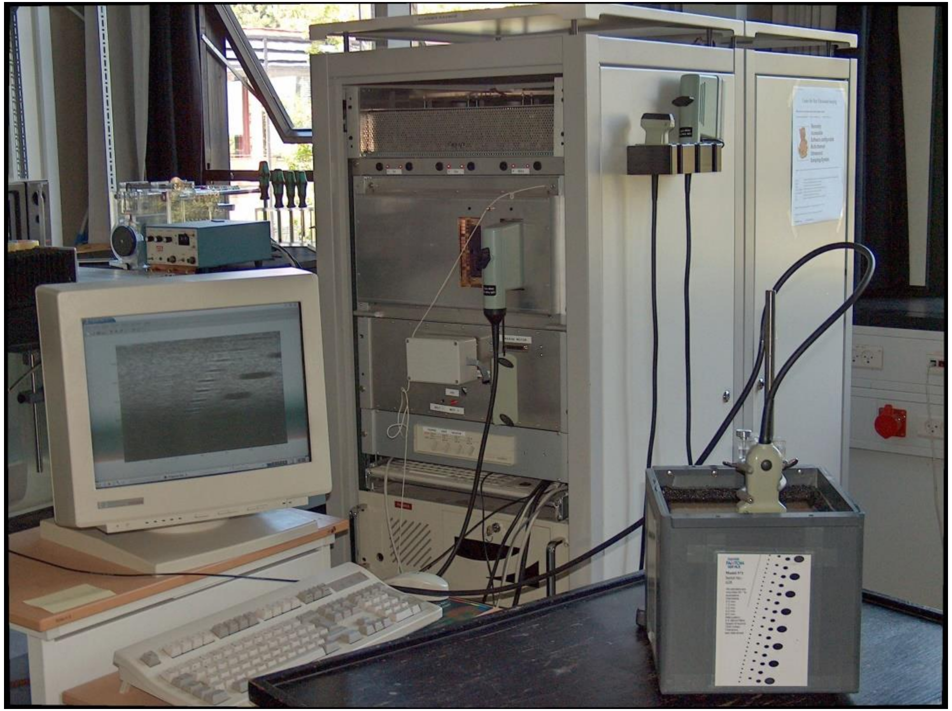
Photo out of paper by Jensen/Mathorhe, 1991



Drawing by J. A. Jensen, 1996

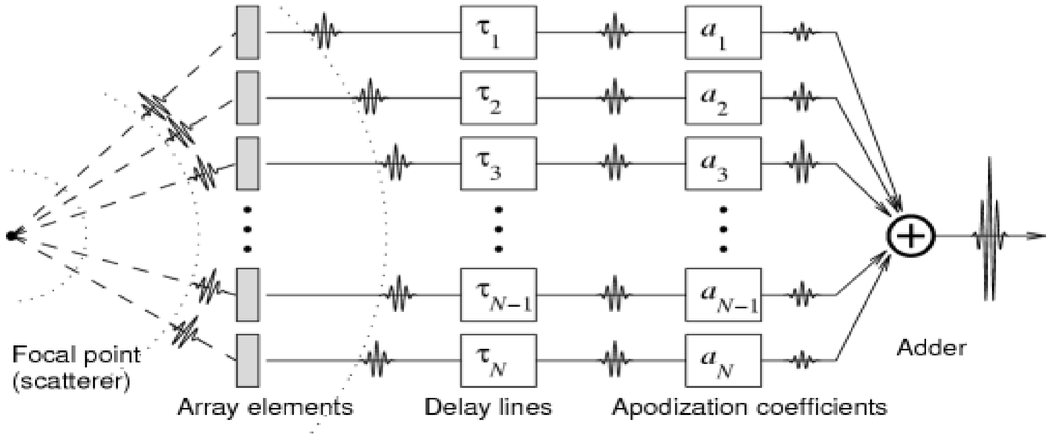
RASMUS (2001)

Remotely
Accessible
Software programmable
Multi-channel
Ultrasound
System



Center for Fast Ultrasound Imaging
Technical University of Denmark

Purpose of RASMUS



- Flexible transmission
- Storage of data for later experimental beamforming
- Real time processing and imaging for orientation

Block diagram

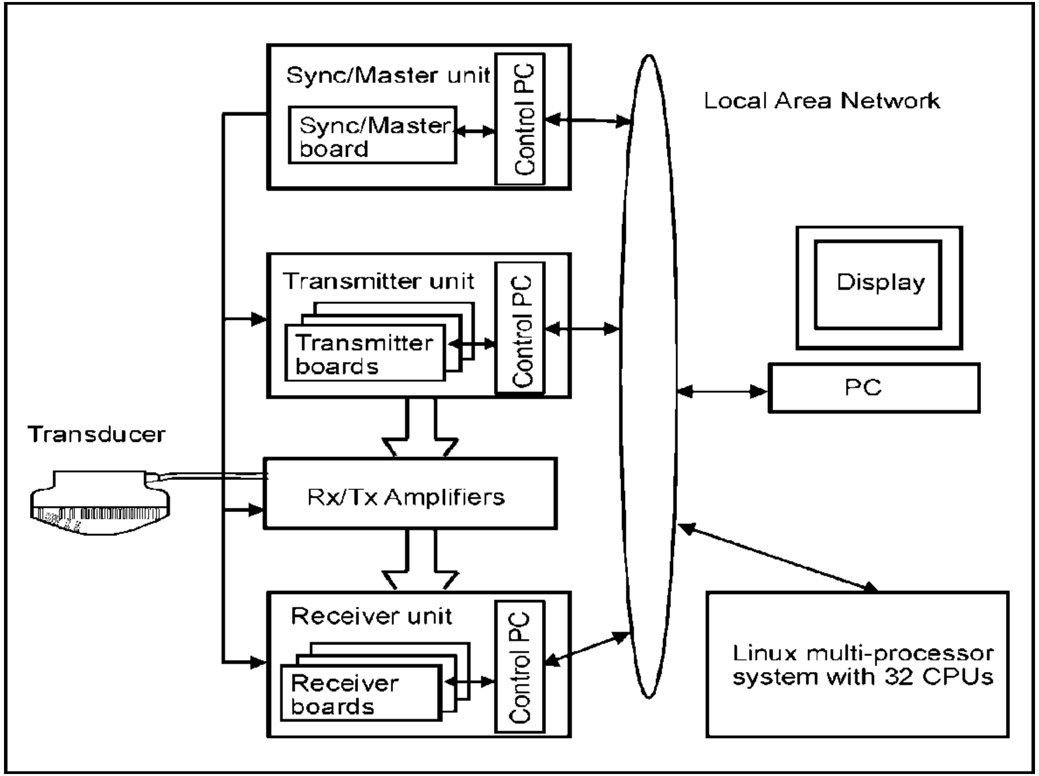


Diagram out of paper by Jensen et al., 1999

Construction

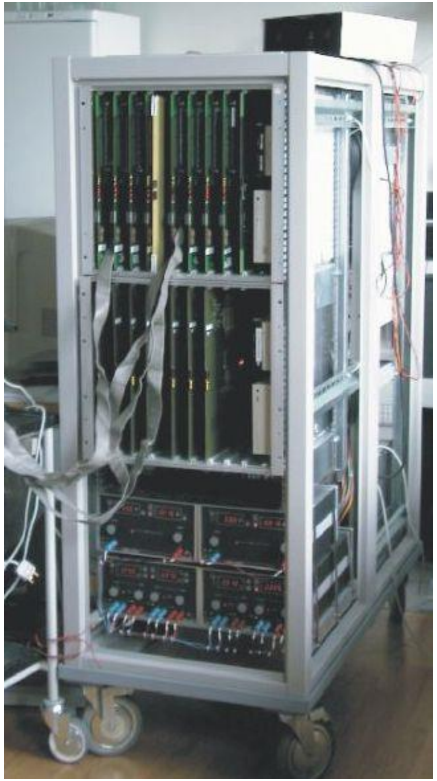
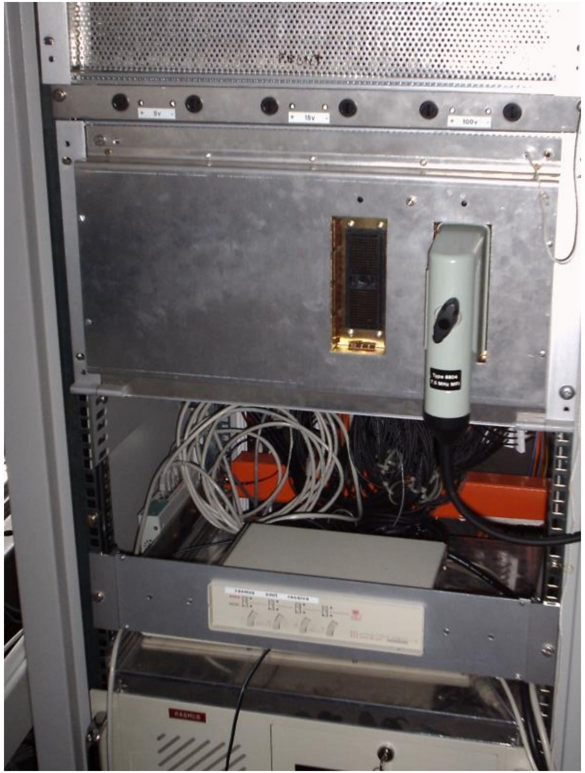
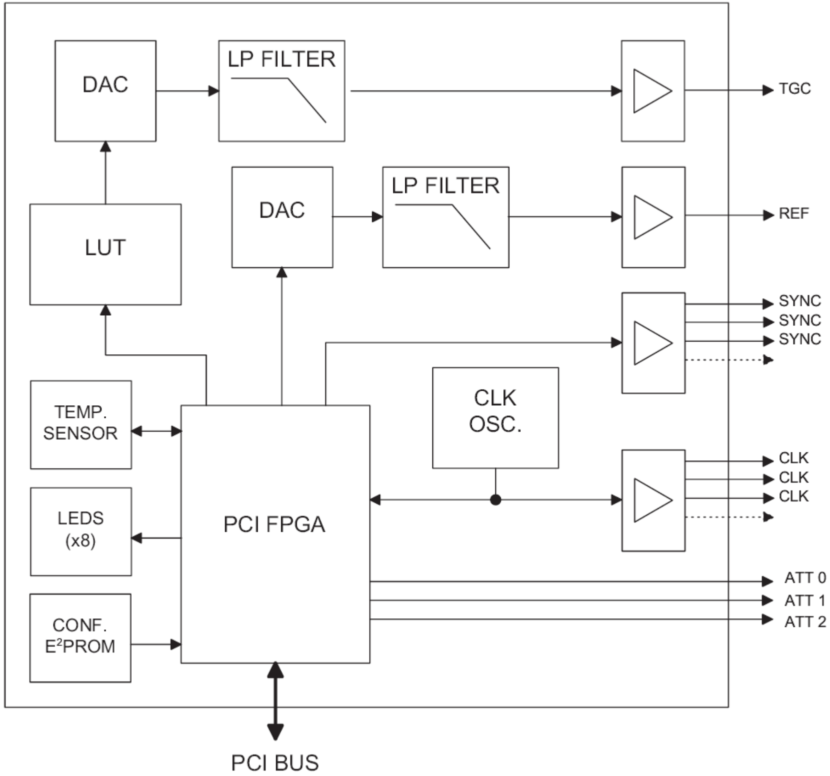


Photo by J.A. Jensen, 2002

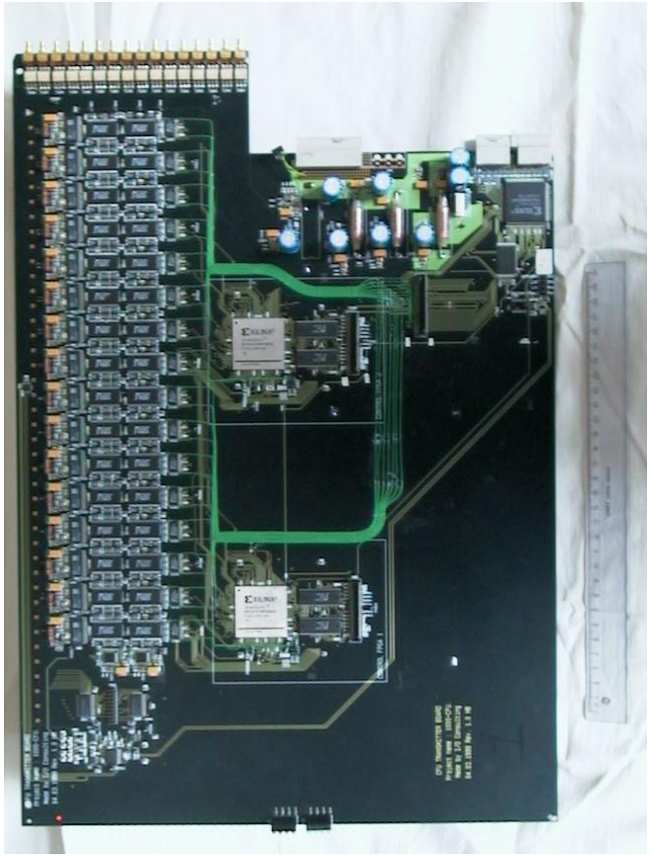


Timing board

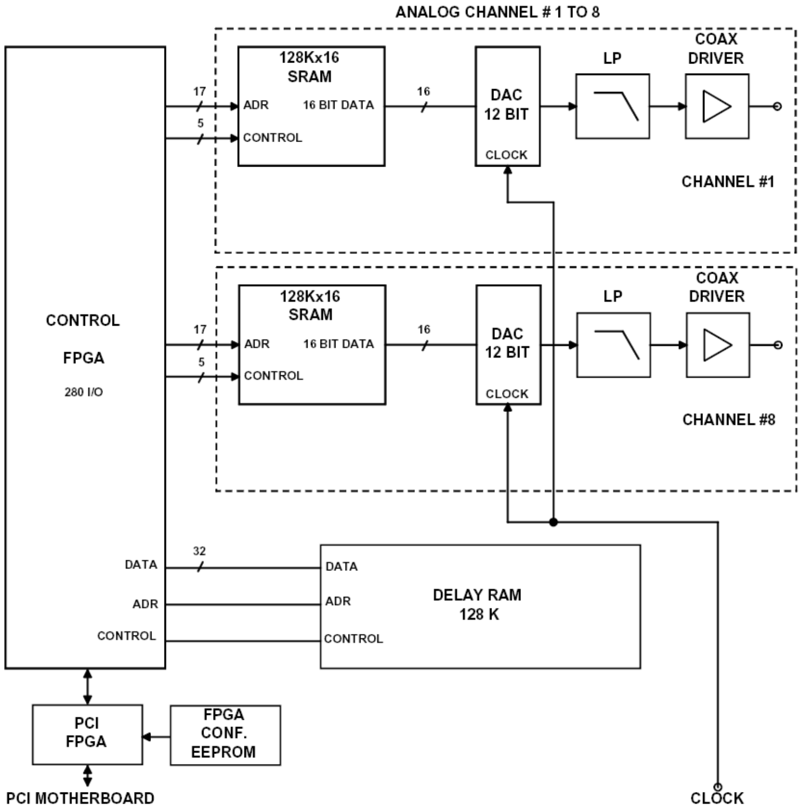


Transmitter boards

- DAC: 40 MHz, 12-bit
- 256 kB per channel waveform RAM
- Independent waveforms for each channel and emission
- 16 channels/board
- 128 channels in total



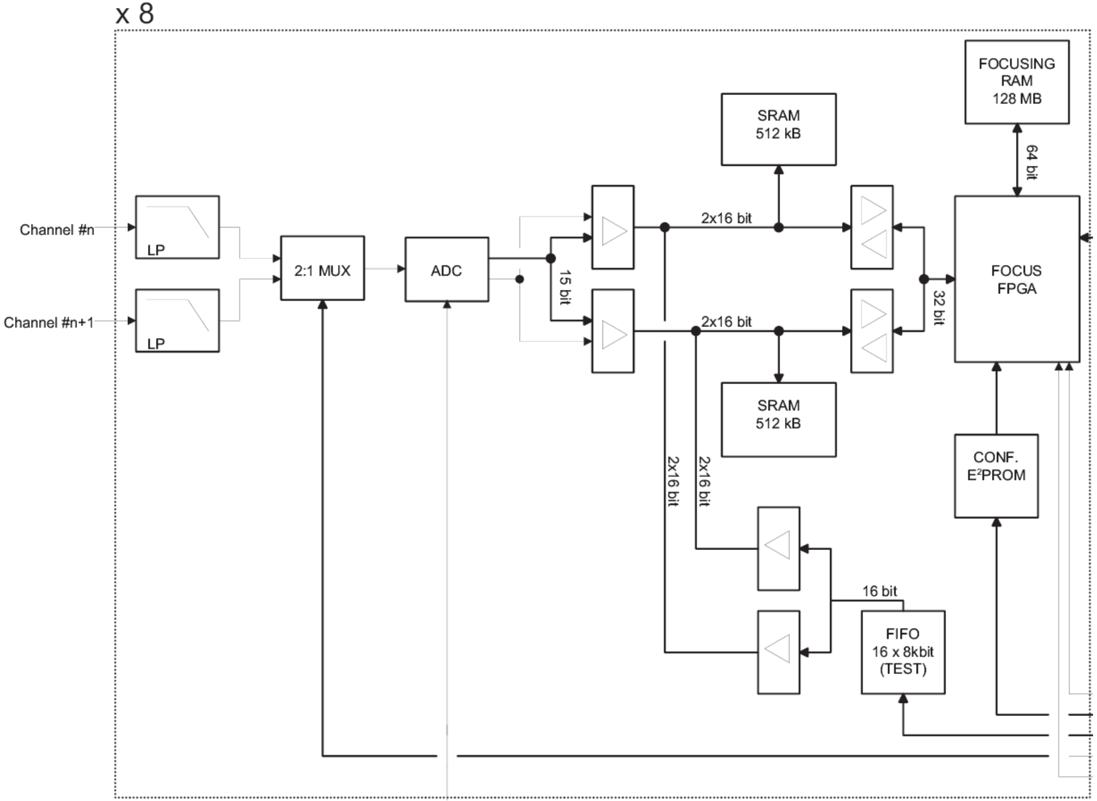
Transmitter boards

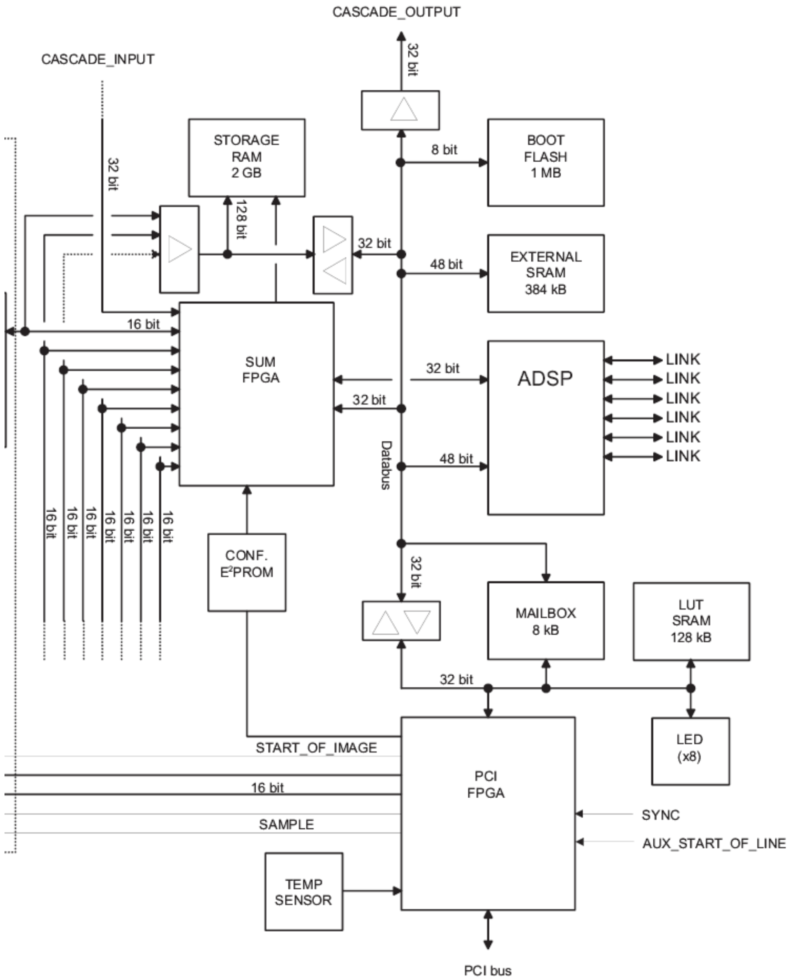


Receiver boards

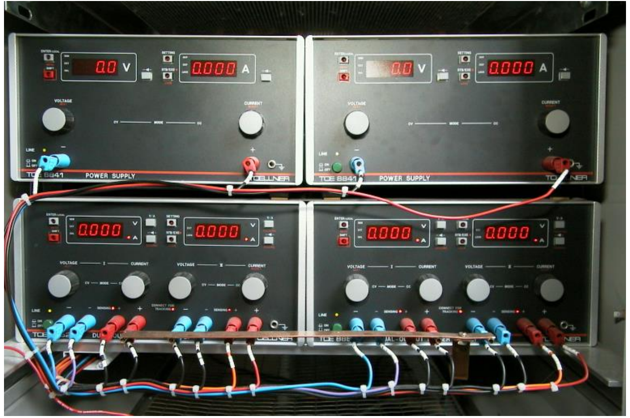
- 8 channels per board
- 2-to-1 multiplexing
- ADC: 40 MHz, 12-bit
- 256 MB RAM per channel (3 seconds of real time data, 2 GB)

Receiver boards




CASCADE_OUTPUT
32 bit
CASCADE_INPUT
32 bit
STORAGE
RAM
2 GB
128 bit
8 bit
BOOT
FLASH
1 MB
32 bit
48 bit
EXTERNAL
SRAM
384 kB
16 bit
SUM
FPGA
32 bit
32 bit
ADSP
LINK
LINK
LINK
LINK
LINK
LINK
48 bit
Databus
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
CONF.
E²PROM
32 bit
MAILBOX
8 kB
LUT
SRAM
128 kB
32 bit
LED
(x8)
START_OF_IMAGE
PCI
FPGA
16 bit
SYNC
AUX_START_OF_LINE
SAMPLE
TEMP
SENSOR
PCI bus

Power supplies



-50 V

Start these last and together, stop first

+ 50 V

+5 V, -5 V

+15 V, -15 V

Outline-software

- Organization
- Commands
  - Initialization and closing
  - Setup - general commands
  - Setup - timing board
  - Setup - transmitter
  - Setup - receiver
  - Acquisition
  - Reading data

Software organization

Host PC:

- user sits at it

- runs Matlab

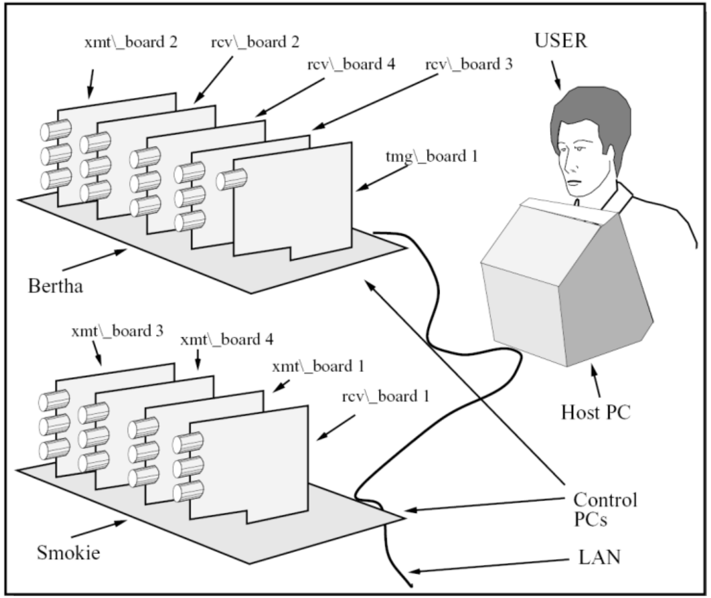
- C library functions called from Matlab

Control PC:

- contains RASMUS boards

- runs drivers

- runs execution server ”sys_master_ctrl”



Initialization and closing

sys_init([*file_name*, [*show_logo*,[*interrupt*]]])

- Uses /home/*username*/.syslib by default

sys_end

- Releases the command server for other users and the memory used by Matlab

sys_abort

- Stops the command server

Setup - general

sys_set_param(*parameter_name*, *parameter_value*)
- For now, **c** and $\mathbf{f_s}$ (default 1540 and 40e6)

sys_set_no_lines(*number_lines* [, *skipped*, *sampled*])

sys_set_sampling_interval(*start_depth*, *end_depth*)

sys_set_fprf(*pulse_repetition_frequency*)

tr_bk8802, tr_bk8804, tr_general, xmt_set_no_samples(*n*)

Setup-timing board

tmg_ref_voltage(*voltage*)

- reference voltage for the TGC amplifiers

tmg_set_attn(*attenuation _code*)

- attenuation of the transmit amplifiers.

tmg_tgc2(*gain_values*)

- 0 to 48 (in dB), 1 value per microsecond

Setup- transmitter

xmt_set_ref_v(*voltage*)

xmt_center_focus(*line_numbers*,*point_coordinates* [, *frame_no*])

xmt_focus(*line_numbers*,*point_coordinates* [, *frame_no*])

xmt_excitation(*samples_normalized*)

xmt_apodization(*line_no*, *apodization* [, *frame_no*])

xmt_mode(*continuous_mode*, *use_external_trigger*)

Setup-receiver

rcv_center_focus(*line_numbers*, *point_coordinates*)

rcv_focus(*line_no*, *switch_pos*, *times*, *focal_points*)

rcv_dynamic_focus (*line_no*, *switch_pos*, *time*, *angle_xz*, *angle_yz*)

rcv_apodization (*line_no*, *times*, *values*)

rcv_mode(...)

Acquisition

tmg_measure(*no_images*)

Reading data

rcv_get_current_image(*brd_no*)

rcv_set_current_image(*offset*, *relative*)

rcv_storage_read_sampled(*channel_numbers*, *image_no*, *line_no*, *to_double*)

rcv_storage_read_summed(*board_no*, *image_no*, *lines_no*, *’class_name’*)

2001 - Rasmine

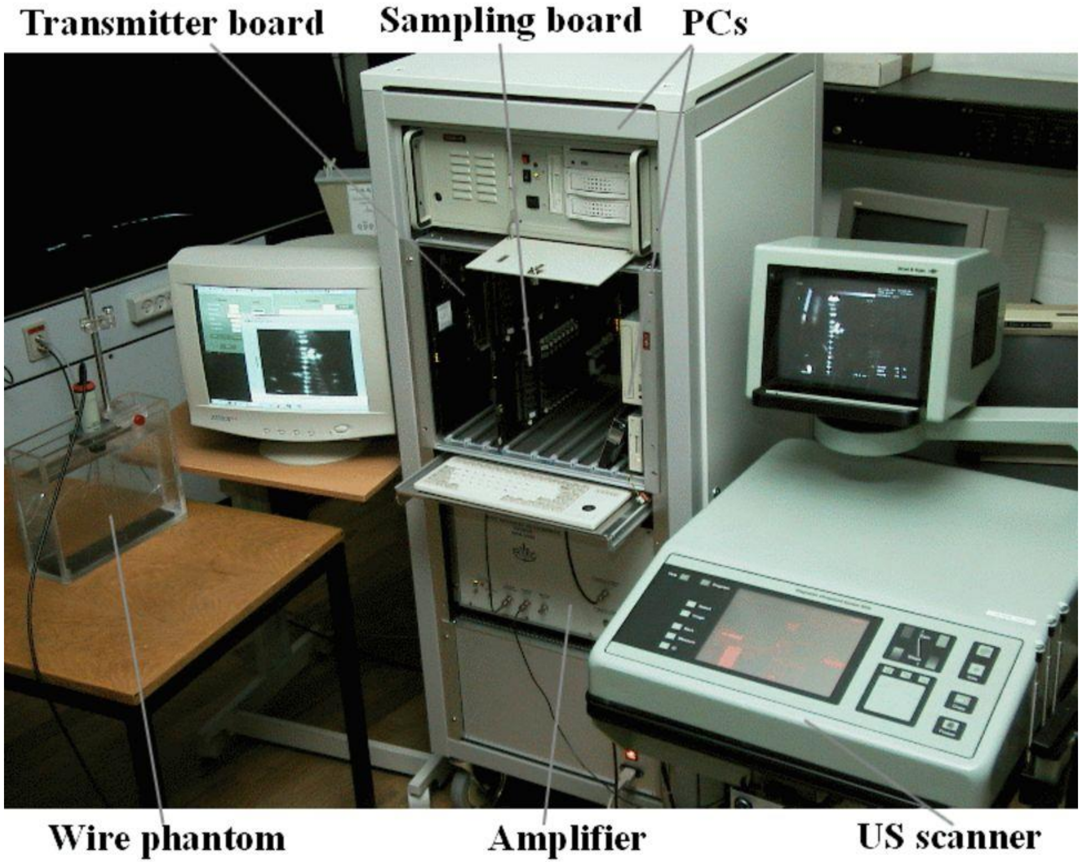
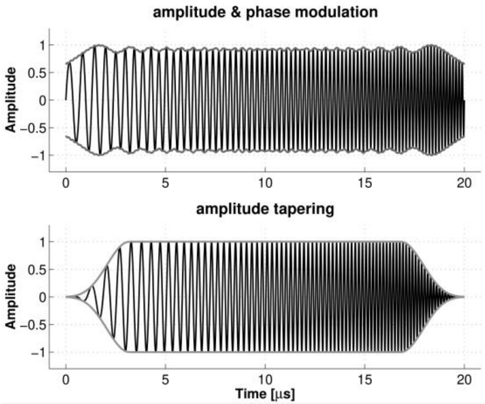
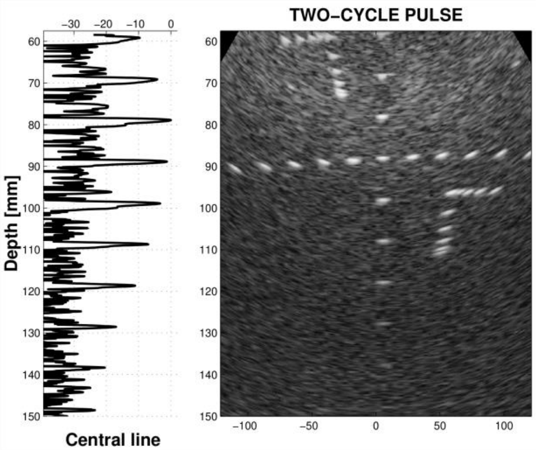


Photo by Thanassis Misaridis,

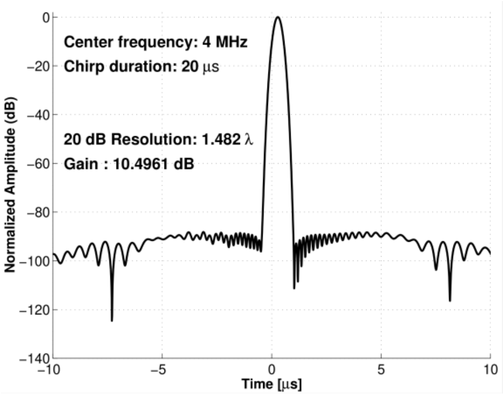
2001

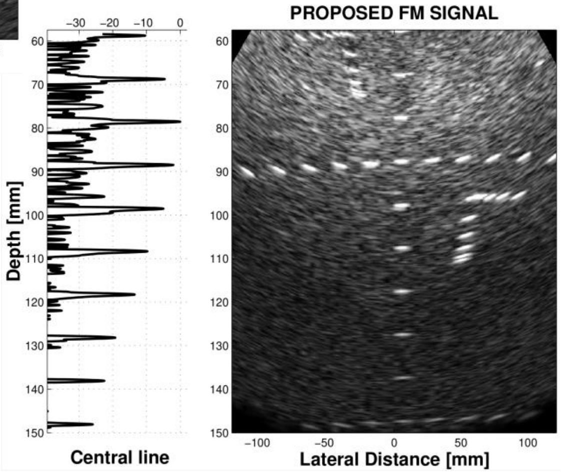
Rasmine - results





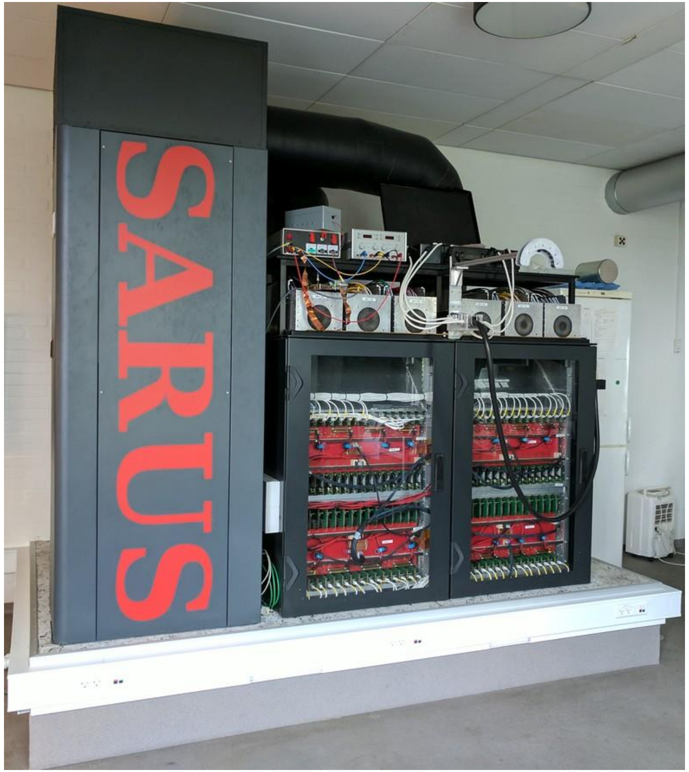
Research by T. Misaridis (1999,2000)





SARUS (2010)

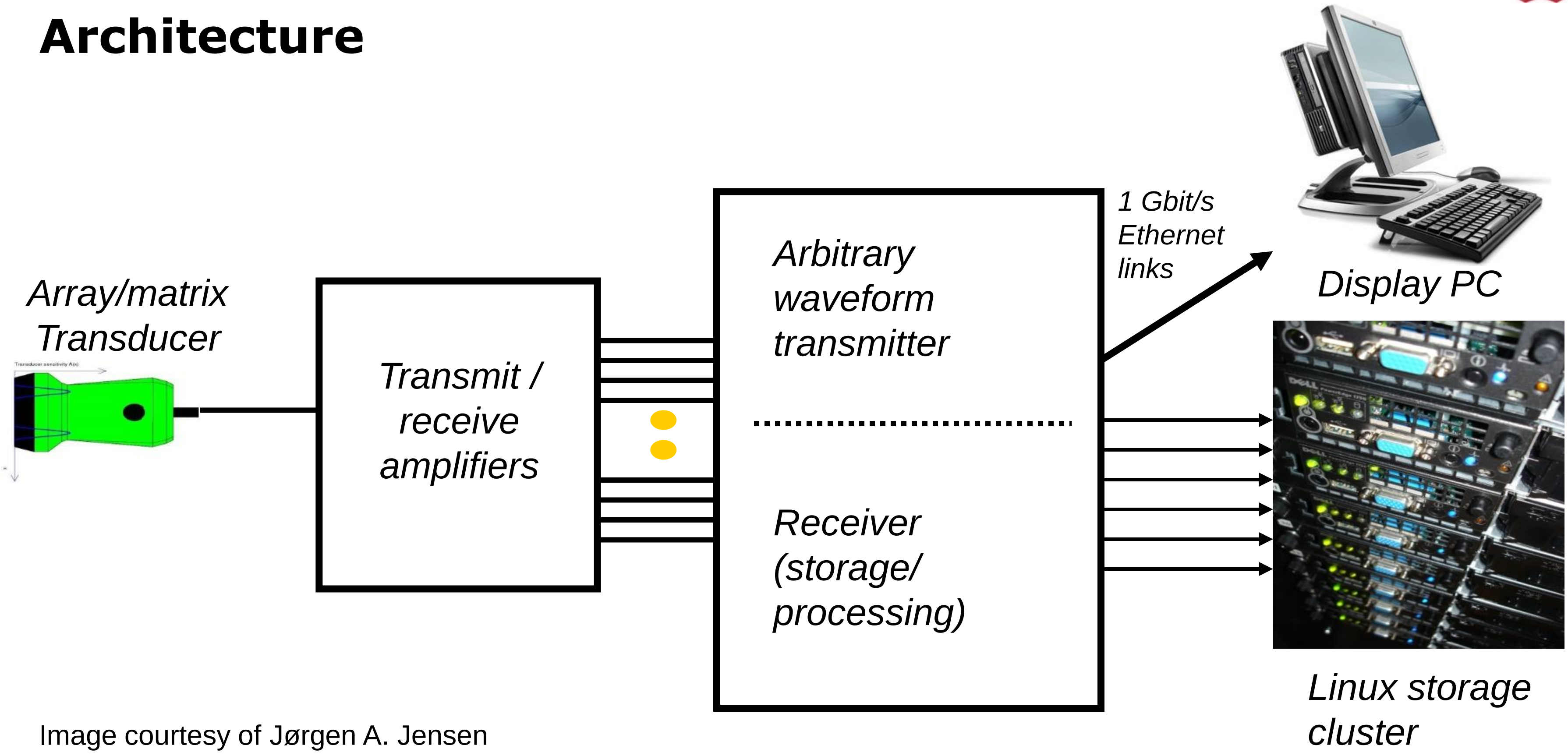
Synthetic
Aperture
Real–time
Ultrasound imaging
System



Purpose of SARUS

- It is an experimental ultrasound imaging system:
  - Flexible transmit side – 1024 independent channels, up to 4096 samples at 70 MHz, up to 8192 different excitations per channel
  - Flexible receive side – selective sampling on 0 to 16 channels per board (0, 4 or higher even numbers), 1024 channels in total, 1 second continuous sampling at 70 MHz
  - Real-time preview / navigation capability - also using SA imaging
  - Transportable

Architecture



Array/matrix Transducer

Transmit / receive amplifiers

Arbitrary waveform transmitter

Receiver (storage/ processing)

1 Gbit/s Ethernet links

Display PC

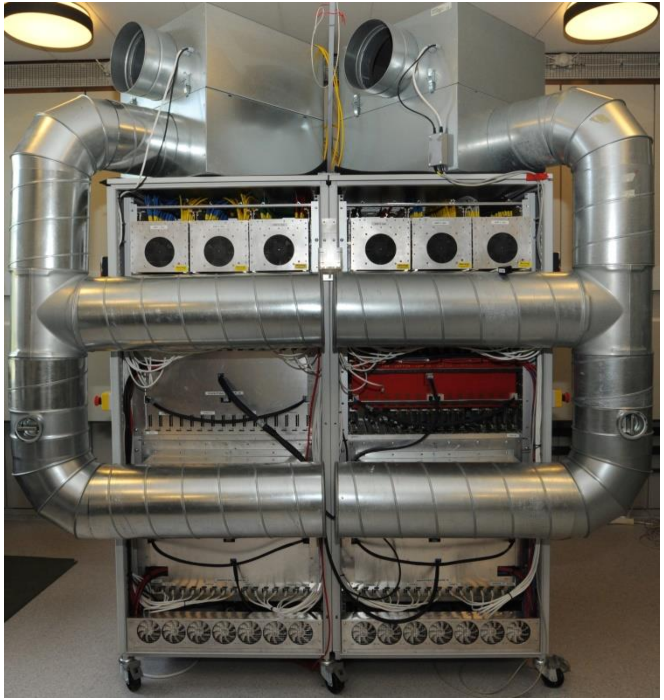
Linux storage cluster

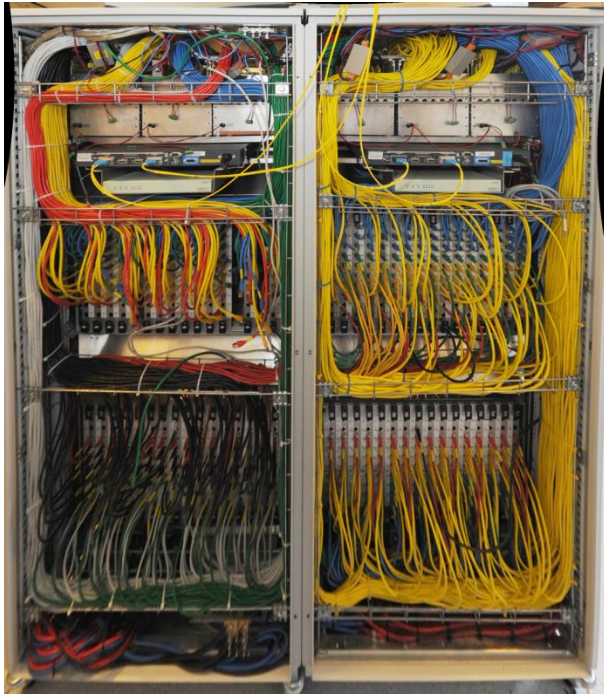
Image courtesy of Jørgen A. Jensen

Hardware components

- Digital acquisition / processing boards
  - 64 boards x 16 channels, 1 board is timing ctrl
  - Distributed in 4 racks / 2 cabinets
- Transmit / receive amplifiers
  - 128 amplifier boards in 6 boxes, up to 24 brd. per box
- 6 B-K transducer connectors (5 x 192 ch. and 1 x 64 ch.)
- Cabling – 512 cables

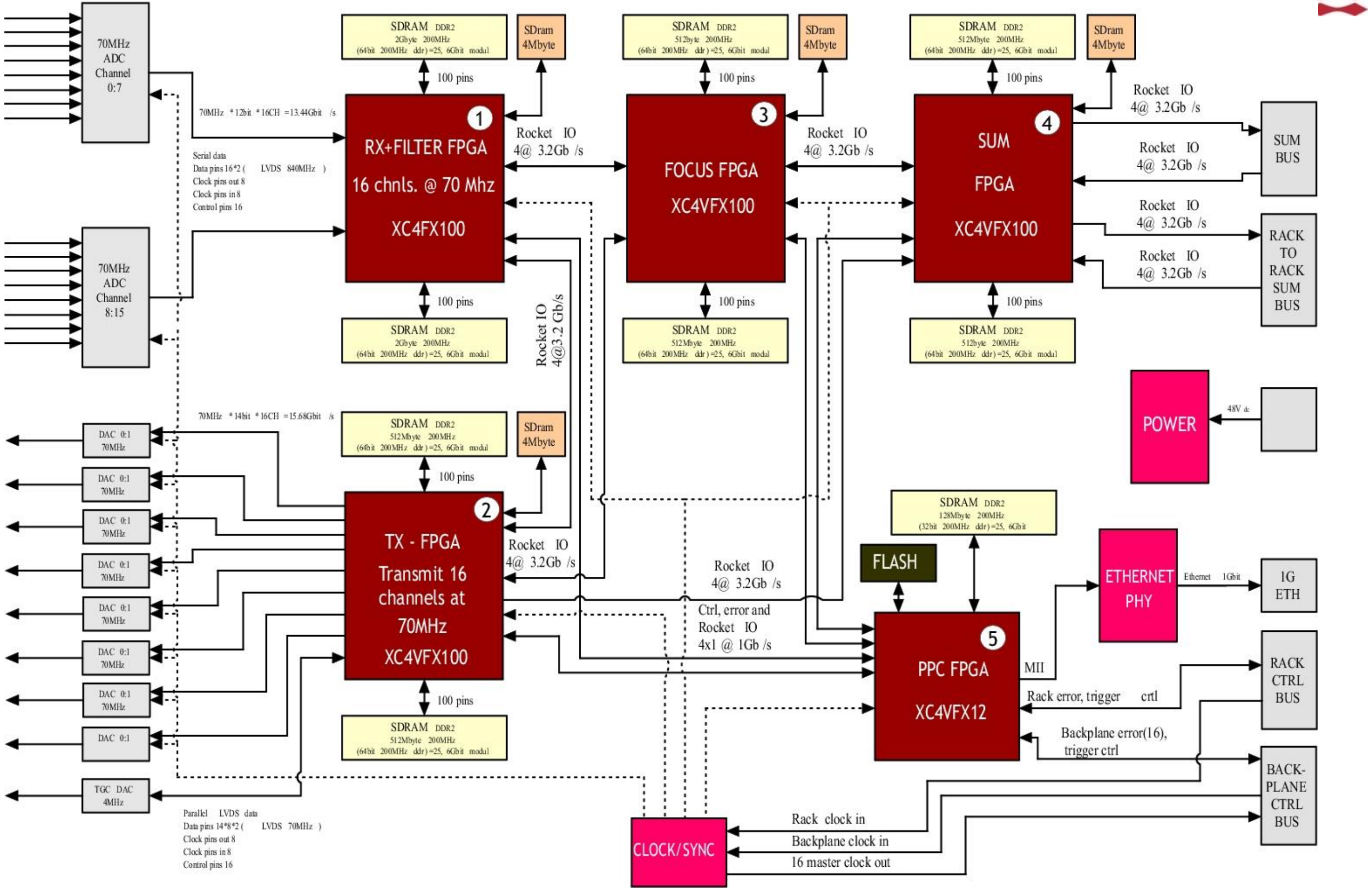
Initial cooling setup



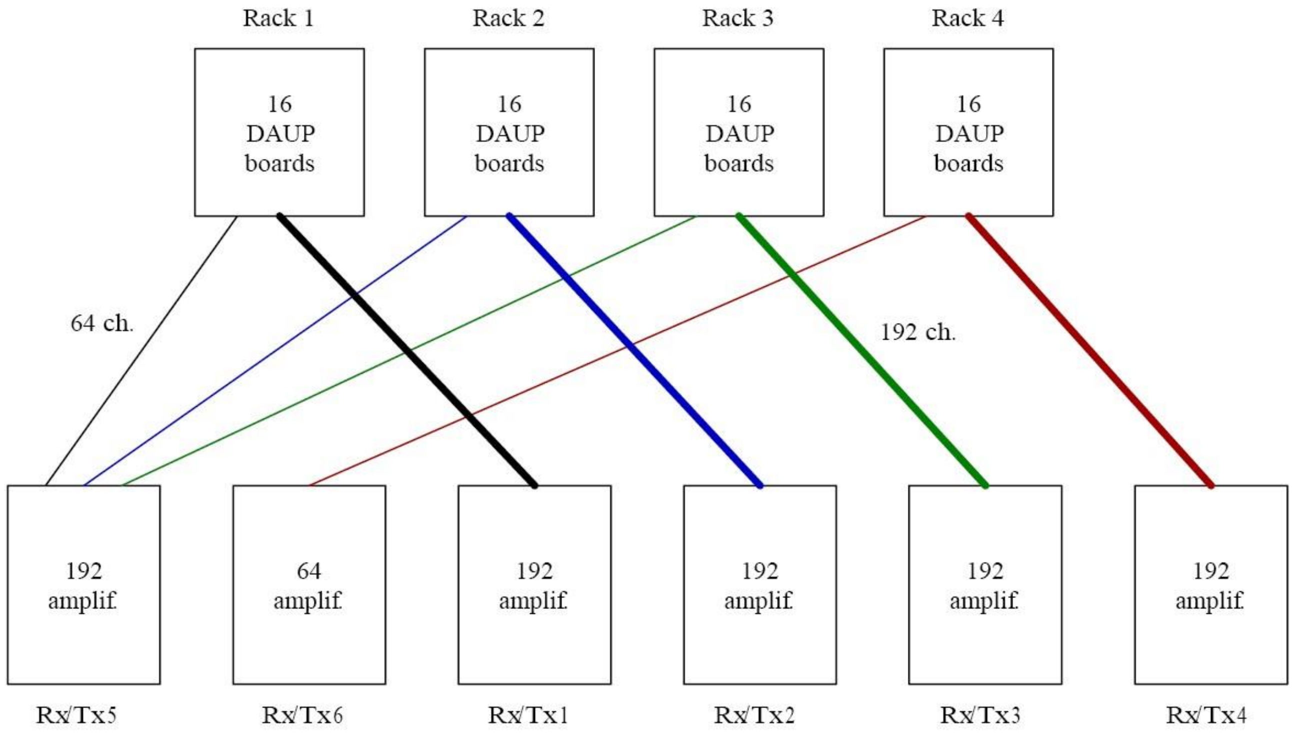


Photos by M. F. Rasmussen, 2012

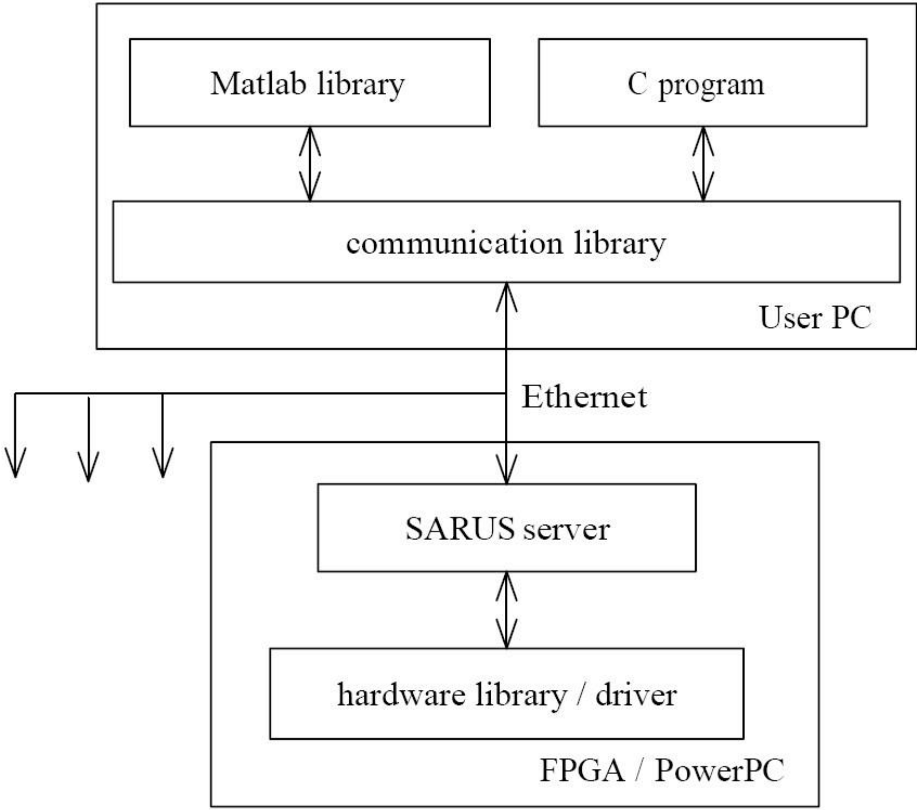
DAUP board


70MHz
ADC
Channel
0:7
70MHz
ADC
Channel
8:15
70MHz * 12bit * 16CH = 13.44Gbit /s
Serial data
Data pins 16*2 (LVDS 840MHz)
Clock pins out 8
Clock pins in 8
Control pins 16
SDRAM DDR2
2Gbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
SDram
4Mbyte
100 pins
1
RX+FILTER FPGA
16 chnls. @ 70 Mhz
XC4FX100
100 pins
SDRAM DDR2
2Gbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
Rocket IO
4@ 3.2Gb /s
Rocket IO
4@3.2 Gb/s
SDRAM DDR2
512byte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
SDram
4Mbyte
100 pins
3
FOCUS FPGA
XC4VFX100
100 pins
SDRAM DDR2
512Mbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
Rocket IO
4@ 3.2Gb /s
SDRAM DDR2
512Mbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
SDram
4Mbyte
100 pins
4
SUM
FPGA
XC4VFX100
100 pins
SDRAM DDR2
512byte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
Rocket IO
4@ 3.2Gb /s
Rocket IO
4@ 3.2Gb /s
SUM
BUS
Rocket IO
4@ 3.2Gb /s
Rocket IO
4@ 3.2Gb /s
RACK
TO
RACK
SUM
BUS
POWER
48V dc
70MHz * 14bit * 16CH = 15.68Gbit /s
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
70MHz
DAC 0:1
TGC DAC
4MHz
SDRAM DDR2
512Mbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
SDram
4Mbyte
100 pins
2
TX - FPGA
Transmit 16
channels at
70MHz
XC4VFX100
100 pins
SDRAM DDR2
512Mbyte 200MHz
(64bit 200MHz ddr) =25, 6Gbit modul
Rocket IO
4@ 3.2Gb /s
Parallel LVDS data
Data pins 14*8*2 (LVDS 70MHz)
Clock pins out 8
Clock pins in 8
Control pins 16
Rocket IO
4@ 3.2Gb /s
Ctrl, error and
Rocket IO
4x1 @ 1Gb /s
SDRAM DDR2
128Mbyte 200MHz
(32bit 200MHz ddr) =25, 6Gbit
FLASH
5
PPC FPGA
XC4VFX12
MII
ETHERNET
PHY
Ethernet 1Gbit
1G
ETH
Rack error, trigger crtl
RACK
CTRL
BUS
Backplane error(16),
trigger ctrl
BACK-
PLANE
CTRL
BUS
CLOCK/SYNC
Rack clock in
Backplane clock in
16 master clock out

Cable connections



Software structure



Concepts/terms

- Image – a pretty picture for display, made of lines
- An emission provides data for one image line, or for a whole low-resolution image in SA imaging.
- Frame – a set of emissions that accomplish the task of providig data for a B-mode image, color flow map, etc.
- A sequence is made of frames in a chain

  /// nowadays, people call a frame sequence

General SARUS commands

- *sarus_init(file_name)*
- *sarus_end*
- *sarus_clear*
- *sarus_reset_fpgas*

Geometry and timing setup commands

- *sarus_use_transducer(xdc_name, serial_num,flags)*
- *sarus_set_speed_of_sound(c)*
- *sarus_create_frame(no_emissions[,...])*
- *sarus_set_tprf(tprf_array)*
- *sarus_set_fprf(fprf_scalar)*

Transmitter setup using virtual sources

- *sarus_xmt_define_excitation(vector)*
- *sarus_xmt_define_virtual_source(start_e, end_e, weights, delays, wavetype, prop_dir_focus, use_fine_delay)*
- *sarus_xmt_define_virtual_source_rc(....)*
- *sarus_xmt_set_emission_vs(em, virt_srcs, ha, weights)*

Receiver setup

- *sarus_set_sampling(emissions, start_d, end_d, elements_store, elements_process)*
- *sarus_set_sampling_rc(emissions, start_d, end_d, elements_store, elements_process)*
- *sarus_set_sampling_times(emissions, start_t, end_t, elements_store, elements_process)*
- *sarus_tgc(emissions, tgc_vector), 5 / microsec.*
- *sarus_set_decimation(dec_factor,use_avg)*

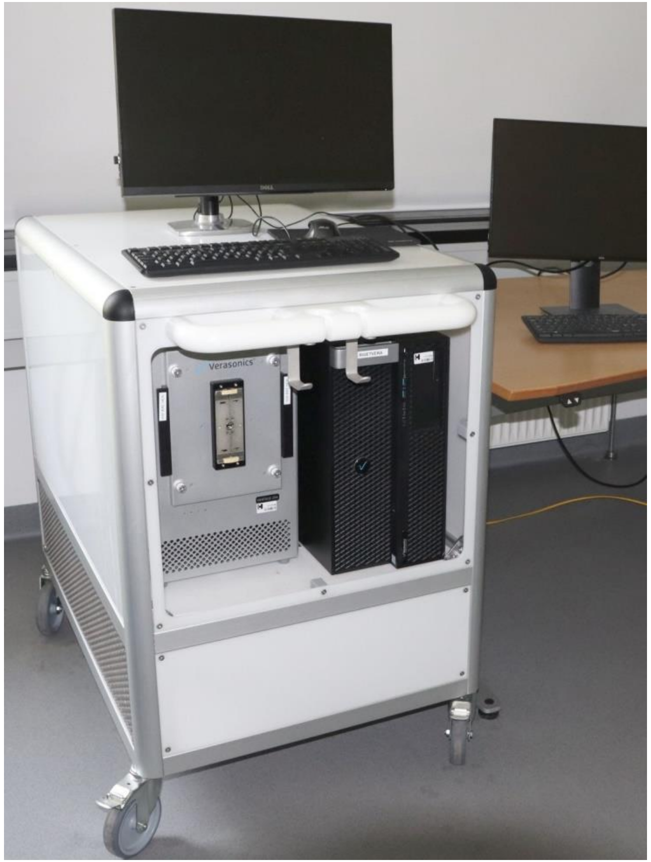
Reading data

- *sarus_read_element_data(elements, frame, em)*
- *sarus_read_frame_data(frame, st_em, no_em...)*
- *sarus_read_single_channel(ch_idx, no_frm...)*

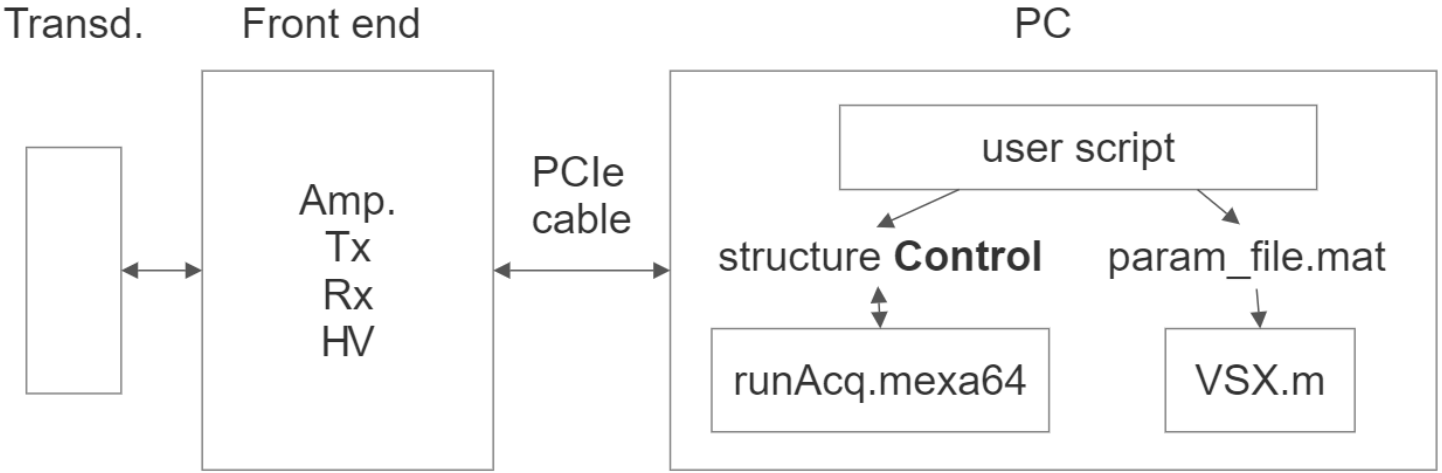
Saving data

- *sarus_set_description_file(file_name)*
- *sarus_set_emission_types(frm_type, em_type,fr)*
- *sarus_set_scan_object(par_name, par_value)*
- *sarus_save_data_set2(no_seq[, path, struct])*
- *sarus_compress_acquisition(path)*

2020 – Vantage 256



Vantage control structure



Vantage setup parameters

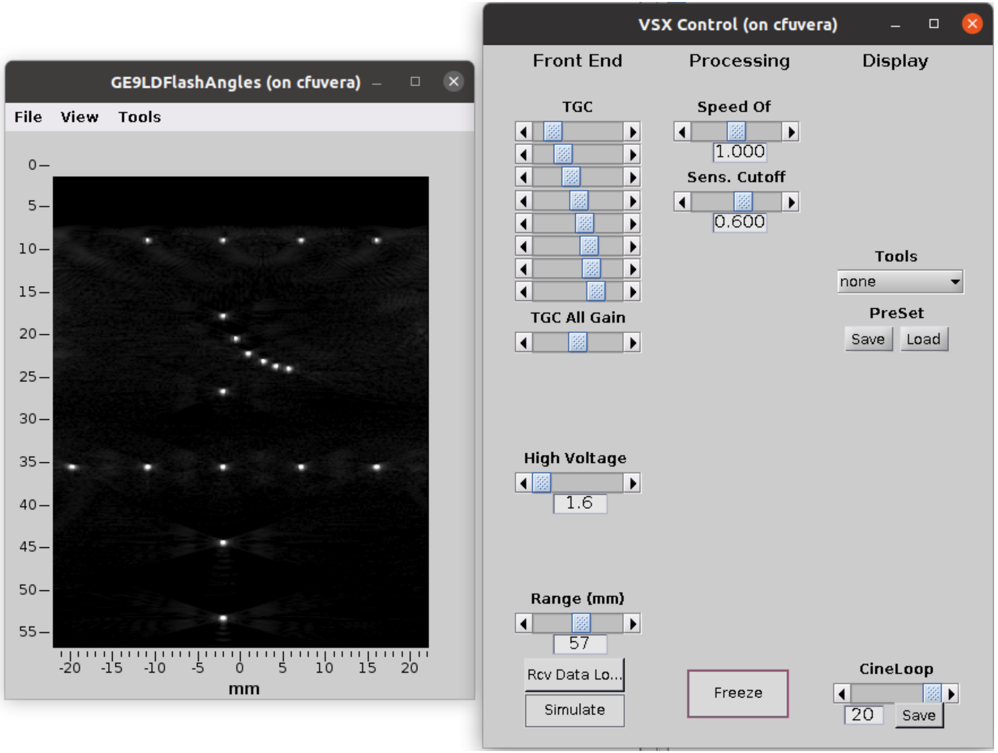
- Resource
- Trans
- TW
- TGC
- TX
- Receive
- Event
- SeqControl
- TPC

- PData
- Media
- Recon
- Process
- UI

Vantage interaction/control

- Structure Control:
  - field Command
  - field Parameters

Vantage GUI (default)



Speed Of: 1.000

Sens. Cutoff: 0.600

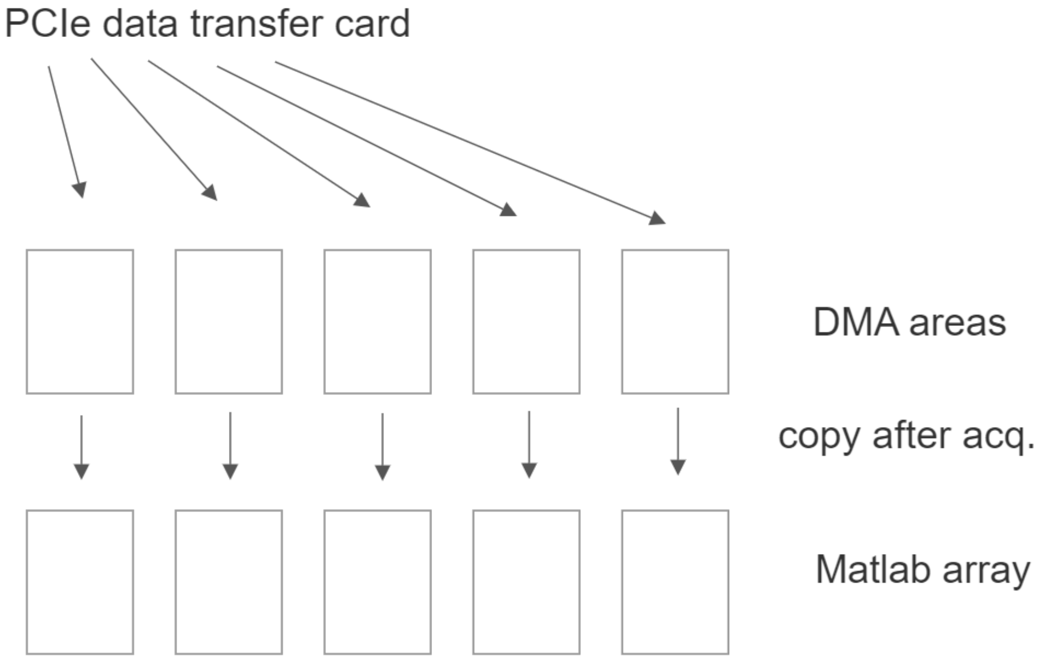
Tools: none

High Voltage: 1.6

Range (mm): 57

CineLoop: 20

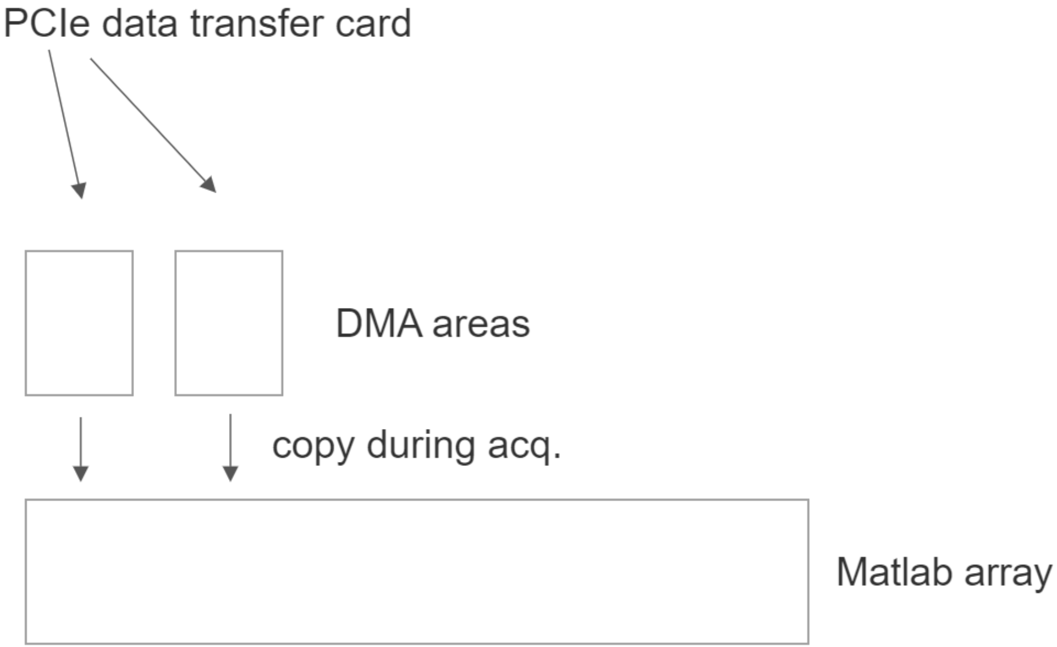
Vantage data transfer mechanism (default)



Vantage system hardware limitations

- 132 000 emissions, at Fprf=5000 gives 26 seconds
- PC RAM utilization < 50 % with default data transfer mechanism
- DMA transfer size > 64 MB for performance, 2GB max (at CFU: 1.7 GB),
- 3-level transmit
- Tx apodization result not visible
- Tx waveform synthesis has discrete center frequency values
- PC RAM allocation takes 1 sec/GB
- The PC runs a non-real-time OS, GUI operations eat time, disturb acq.

Vantage data transfer mechanism (CFU)



PCIe data transfer card

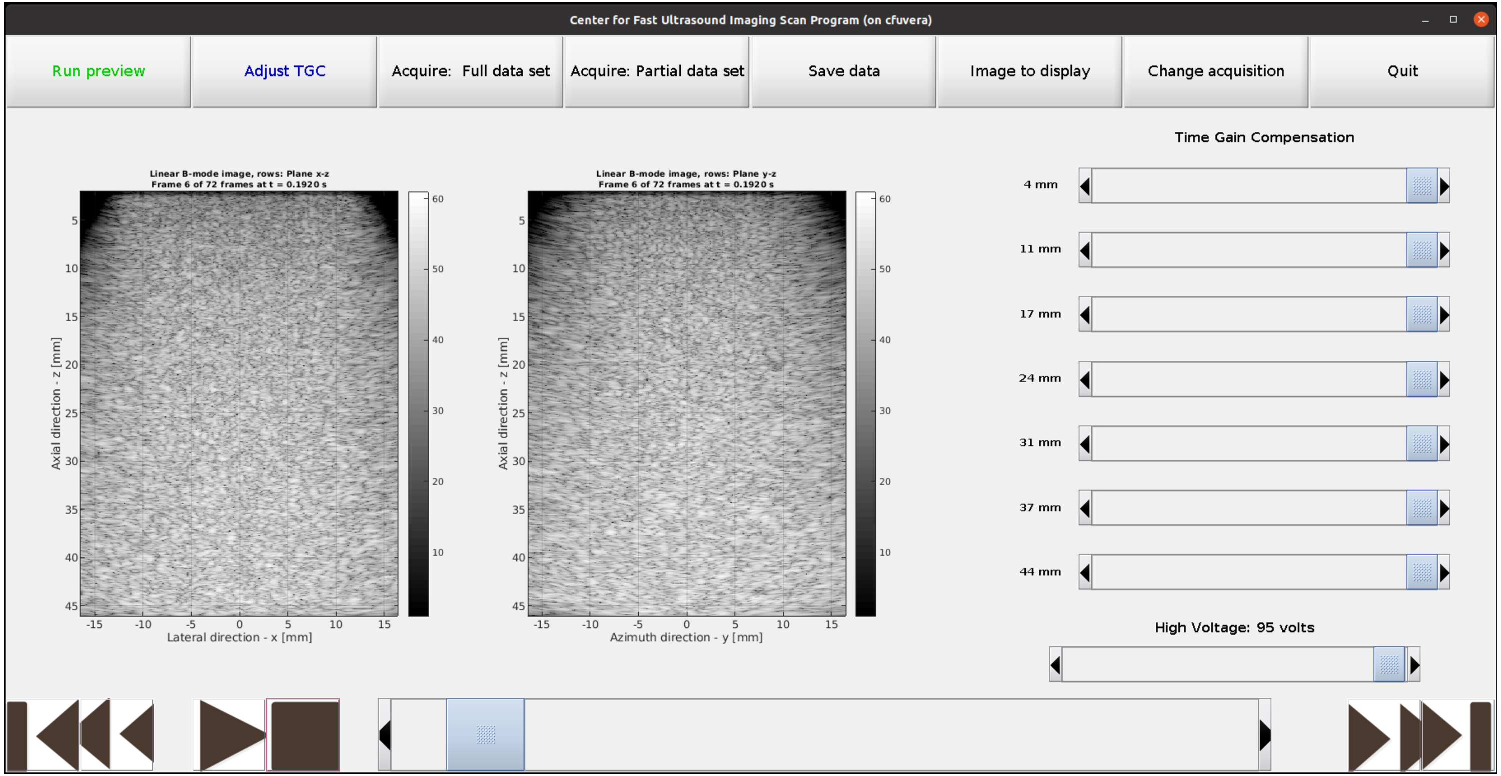
DMA areas

copy during acq.

Matlab array

By idea of Ron Daigle (Verasonics)

CFU_scan



Center for Fast Ultrasound Imaging Scan Program (on cfuvera)
Run preview
Adjust TGC
Acquire: Full data set
Acquire: Partial data set
Save data
Image to display
Change acquisition
Quit
Linear B-mode image, rows: Plane x-z
Frame 6 of 72 frames at t = 0.1920 s
Axial direction - z [mm]
Lateral direction - x [mm]
Linear B-mode image, rows: Plane y-z
Frame 6 of 72 frames at t = 0.1920 s
Axial direction - z [mm]
Azimuth direction - y [mm]
5
10
15
20
25
30
35
40
45
-15
-10
-5
0
5
10
15
60
50
40
30
20
10
Time Gain Compensation
4 mm
11 mm
17 mm
24 mm
31 mm
37 mm
44 mm
High Voltage: 95 volts

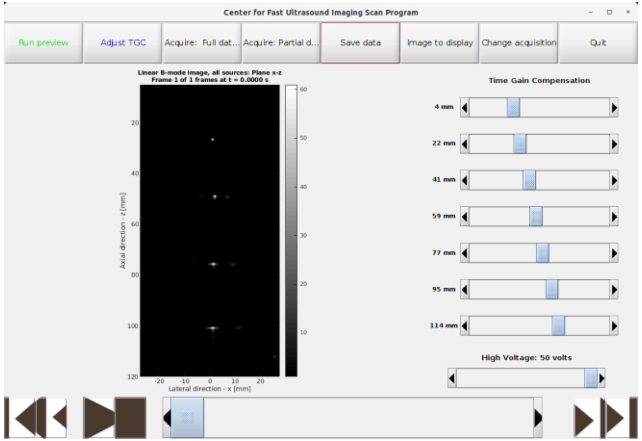
Scanner parameters

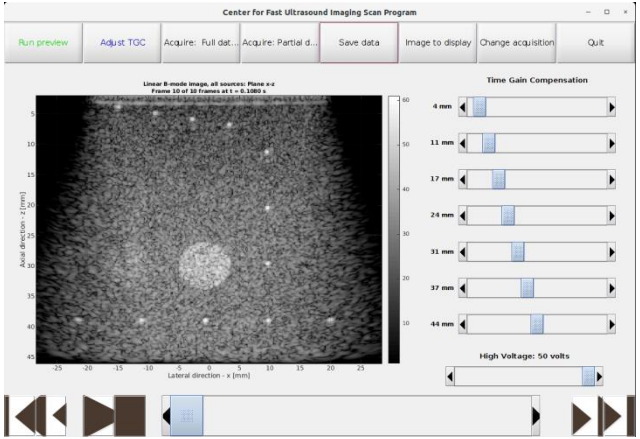
Scanner	**2-ch. system**	**RASMUS**	**SARUS**	**Vantage 256**	**ULA-OP**
In use since, year	1991	2000	2010	2020	-
Channels	2	128	1024	256 (x4)	356
Fs, MHz	20	40	70	62.5	78
RAM, GB	7-12 MB	16	128	PC*	80
Throughput, GB/s	0.04	5.12	143.36	3.5 (max. 6.6)	40
Sampling time, s	0.17	3.4	0.9	160*	2
Transmit	-	Linear	Linear	3-level	Linear
Preview	No	Yes	Yes	Yes	Yes (USB 3)
Mobile	Yes	Yes	No	Yes	Yes

*Vantage PC config. at CFU: 512 GB RAM

Excercise

- Start CFU_scan
- Perform a scan of a wire (1 frame) and a tissue phantom (10 frames)
- Save the RF data
- Beamform it using your own beamformer
- Display the images with correct axes and dynamic range of 60 dB.





How to extract emission data

To extract emission data, use the function:

[filtered_samples, t_start, rx_fs, elem_positions, vsrc_position, c] =
load_scan_data(*path_data*, *frame_index*, *em_idx*),

where the output is:

- filtered_samples - RF data with matched filter applied
- t_start - start time of the RF data
- rx_fs - sampling frequency of the recorded RF data
- elem_position - element positions [N x 3], containing X, Y and Z
- vsrc_position - position of the virtual source
- c - speed of sound in the phantom