„Inspired by bats“

Binaural obstacle detection implemented in GA144

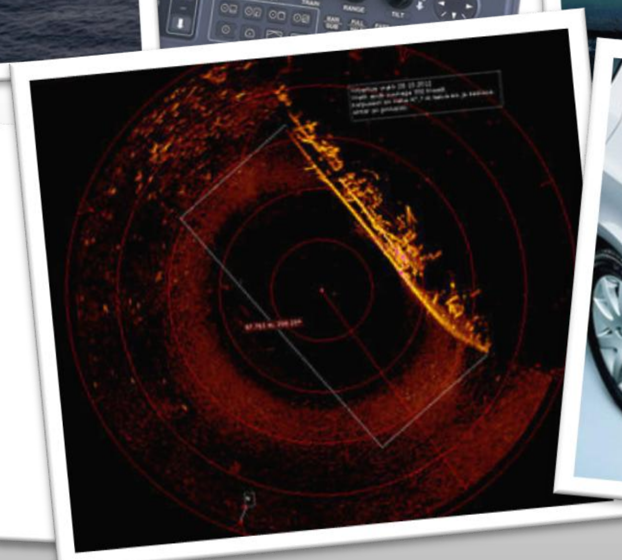
Daniel Kalny
on behalf of GreenArrays

Forth Day 2017

amazing bats...

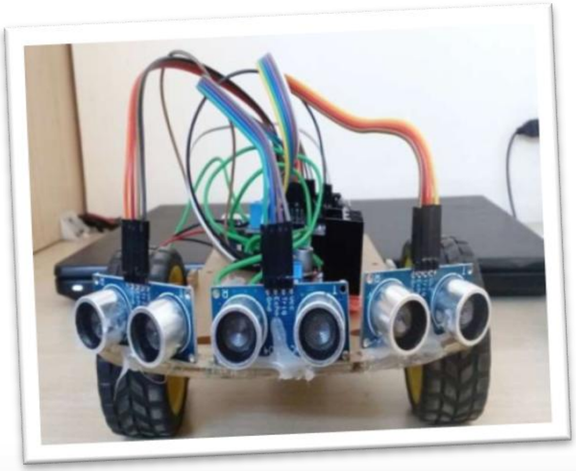
...and their amazing echolocation

sonar - inspired by nature



images: wikimedia, www.furuno.com, author's archive

proximity sensors in robotics



images: robots.net; Omron Adept MobileRobots, LLC; Nomadic Technologies; Denning Branch International Robotics; Real World Interface, Inc.

a creature yet to be discovered

can robots navigate like bats?

- bat echolocation and bahavioral research
  Prof. Cynthia Moss, Johns Hopkins University, MD
  Prof. James A. Simmons, Brown University, RI

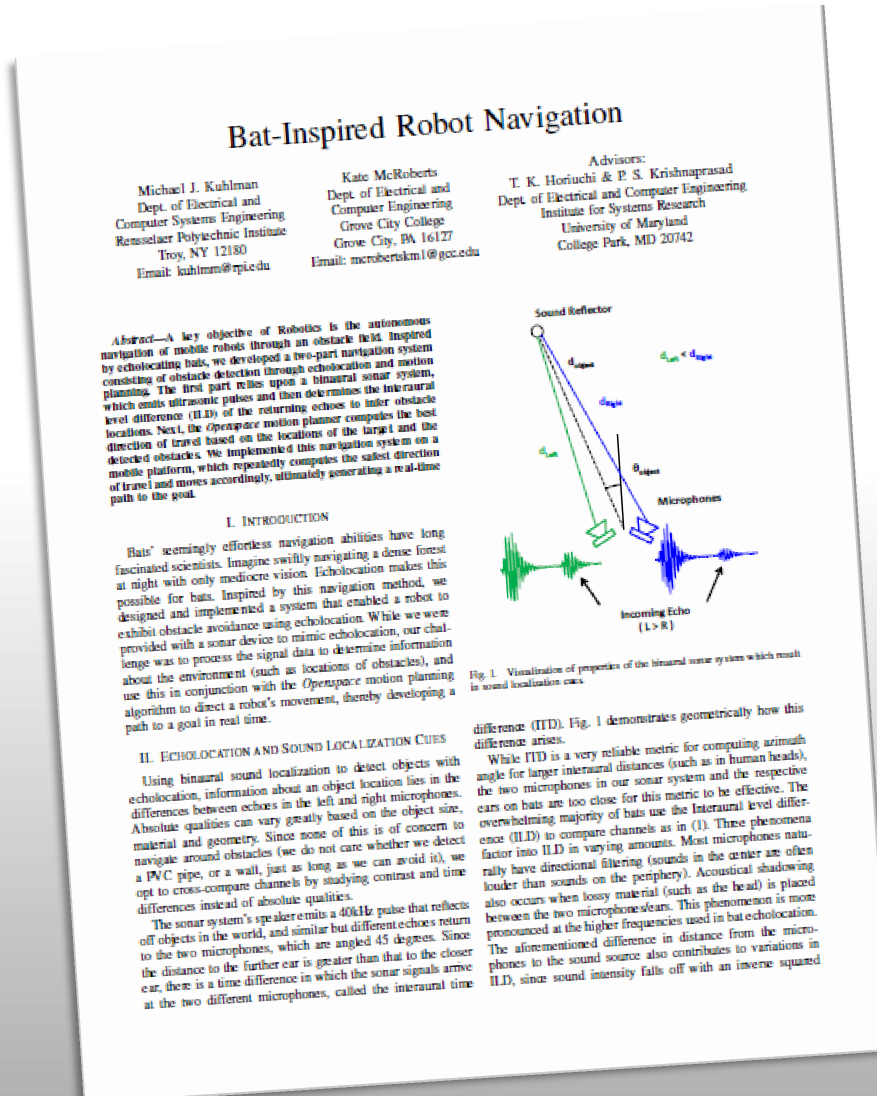
- neuromorphic VLSI implementation
  Dr. Timothy K. Horiuchi, University of Maryland, MD

- biomimetic and bioinspired technology
  Prof. Phillip McKerrow, Wollongong University, Australia
  Prof. Roman Kuc, Yale University, CT
  Prof. John Hallam, University of Southern Denmark
  Dr. Rolf Müller, Virginia Tech, VA

prior work

Michael J. Kuhlman
Kate McRoberts

MERIT Fair 2009
University of Maryland



Bat-Inspired Robot Navigation

Michael J. Kuhlman
Dept. of Electrical and
Computer Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: kuhlmm@rpi.edu

Kate McRoberts
Dept. of Electrical and
Computer Engineering
Grove City College
Grove City, PA 16127
Email: mcrobertskm1@gcc.edu

Advisors:
T. K. Horiuchi & P. S. Krishnaprasad
Dept. of Electrical and Computer Engineering
Institute for Systems Research
University of Maryland
College Park, MD 20742

Abstract—A key objective of Robotics is the autonomous navigation of mobile robots through an obstacle field. Inspired by echolocating bats, we developed a two-part navigation system consisting of obstacle detection through echolocation and motion planning. The first part relies upon a binaural sonar system, which emits ultrasonic pulses and then determines the interaural level difference (ILD) of the returning echoes to infer obstacle locations. Next, the *Openspace* motion planner computes the best direction of travel based on the locations of the target and the detected obstacles. We implemented this navigation system on a mobile platform, which repeatedly computes the safest direction of travel and moves accordingly, ultimately generating a real-time path to the goal.

I. Introduction

Bats' seemingly effortless navigation abilities have long fascinated scientists. Imagine swiftly navigating a dense forest at night with only mediocre vision. Echolocation makes this possible for bats. Inspired by this navigation method, we designed and implemented a system that enabled a robot to exhibit obstacle avoidance using echolocation. While we were provided with a sonar device to mimic echolocation, our challenge was to process the signal data to determine information about the environment (such as locations of obstacles), and use this in conjunction with the *Openspace* motion planning algorithm to direct a robot's movement, thereby developing a path to a goal in real time.

II. Echolocation and Sound Localization Cues

Using binaural sound localization to detect objects with echolocation, information about an object location lies in the differences between echoes in the left and right microphones. Absolute qualities can vary greatly based on the object size, material and geometry. Since none of this is of concern to navigate around obstacles (we do not care whether we detect a PVC pipe, or a wall, just as long as we can avoid it), we opt to cross-compare channels by studying contrast and time differences instead of absolute qualities.

The sonar system's speaker emits a 40kHz pulse that reflects off objects in the world, and similar but different echoes return to the two microphones, which are angled 45 degrees. Since the distance to the further ear is greater than that to the closer ear, there is a time difference in which the sonar signals arrive at the two different microphones, called the interaural time

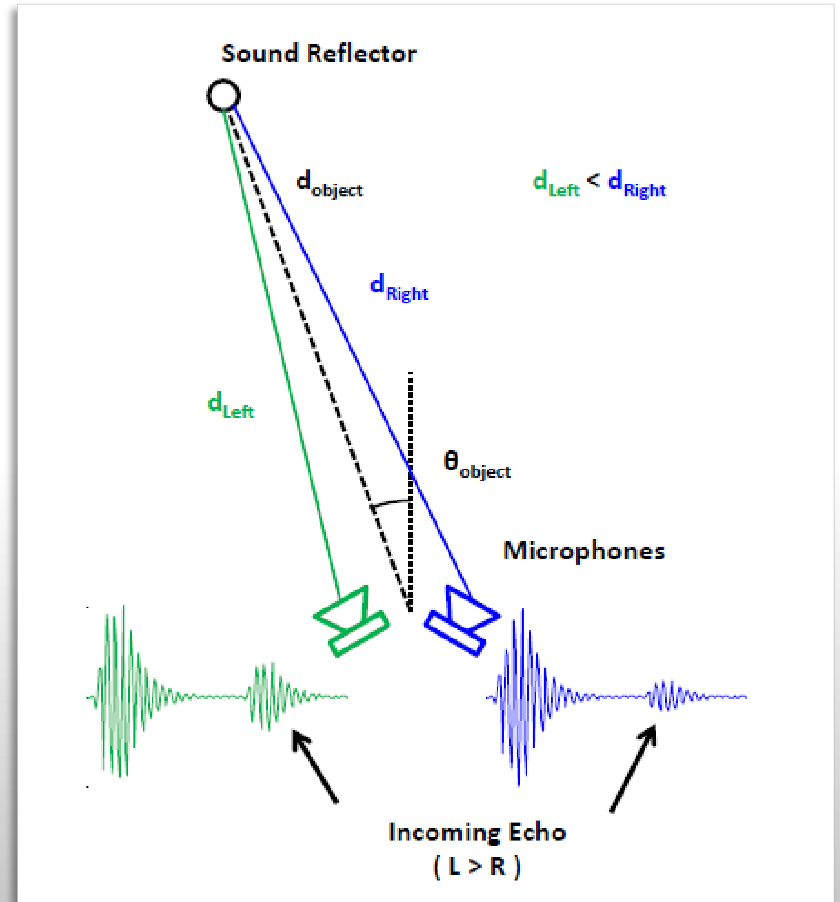
Fig. 1. Visualization of properties of the binaural sonar system which result in sound localization cues.

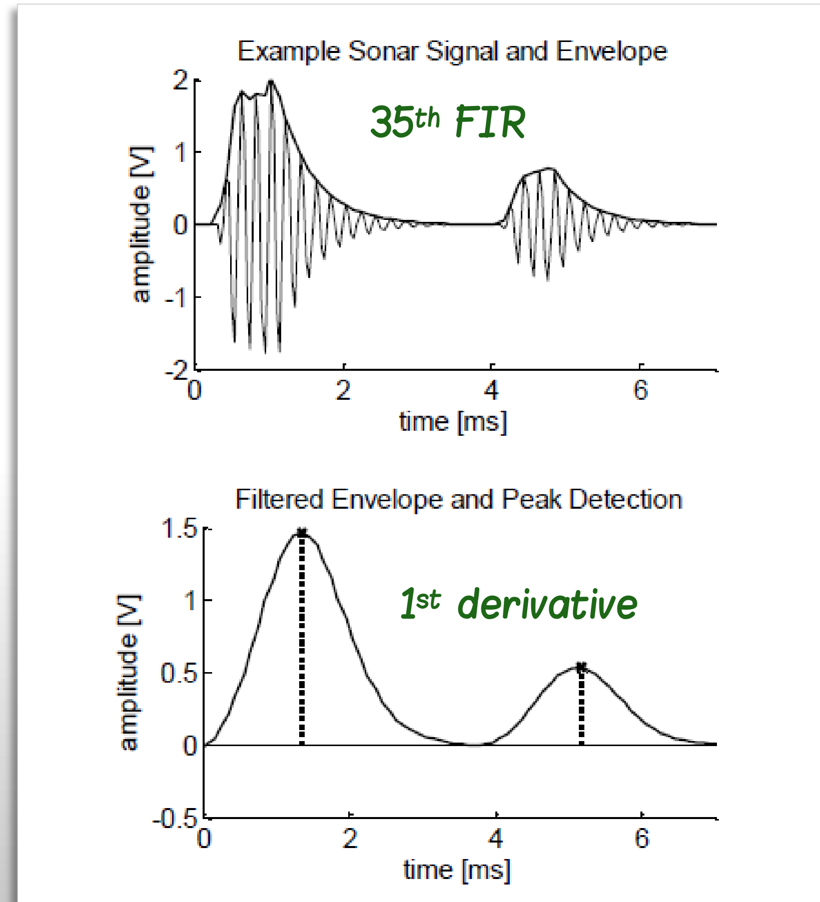
difference (ITD). Fig. 1 demonstrates geometrically how this difference arises.

While ITD is a very reliable metric for computing azimuth angle for larger interaural distances (such as in human heads), the two microphones in our sonar system and the respective ears on bats are too close for this metric to be effective. The overwhelming majority of bats use the Interaural level difference (ILD) to compare channels as in (1). Three phenomena factor into ILD in varying amounts. Most microphones naturally have directional filtering (sounds in the center are often louder than sounds on the periphery). Acoustical shadowing also occurs when lossy material (such as the head) is placed between the two microphones/ears. This phenomenon is more pronounced at the higher frequencies used in bat echolocation. The aforementioned difference in distance from the microphones to the sound source also contributes to variations in ILD, since sound intensity falls off with an inverse squared

www.ece.umd.edu/merit/merit_fair09.htm

prior work





M.J. Kuhlman, K. McRoberts, T.K. Horiuchi and P.S. Krishnaprasad,
"Bat-Inspired Robot Navigation„
Institute for Systems Research, University of Maryland, College Park, MD Tech. Rep. Aug. 2009
with permission

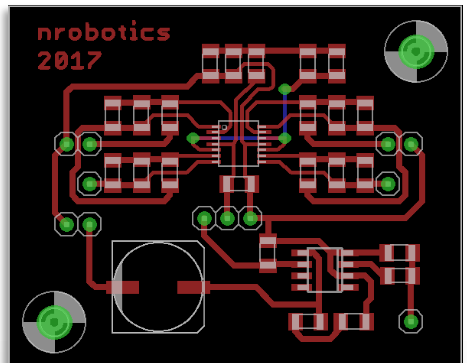
aims of this project

1) implement binaural obstacle detection in GreenArrays chips
2) use a different mathematical approach
   - distributed and parallel processing
   - detection of overlapping echos
   - better noise tolerance
3) develop as a multi-chip application
4) have fun

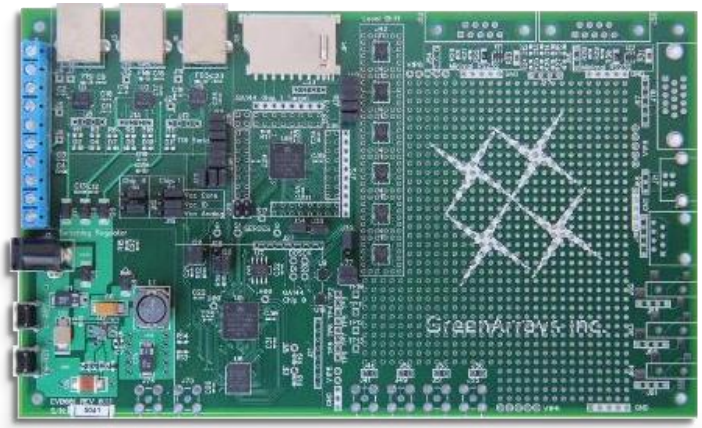
HARDWARE

setup

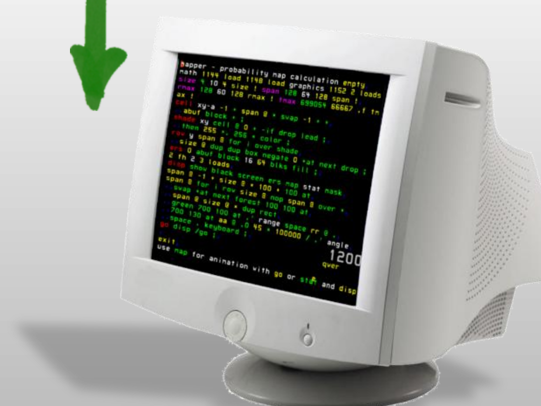
analog front end module



GA eval board

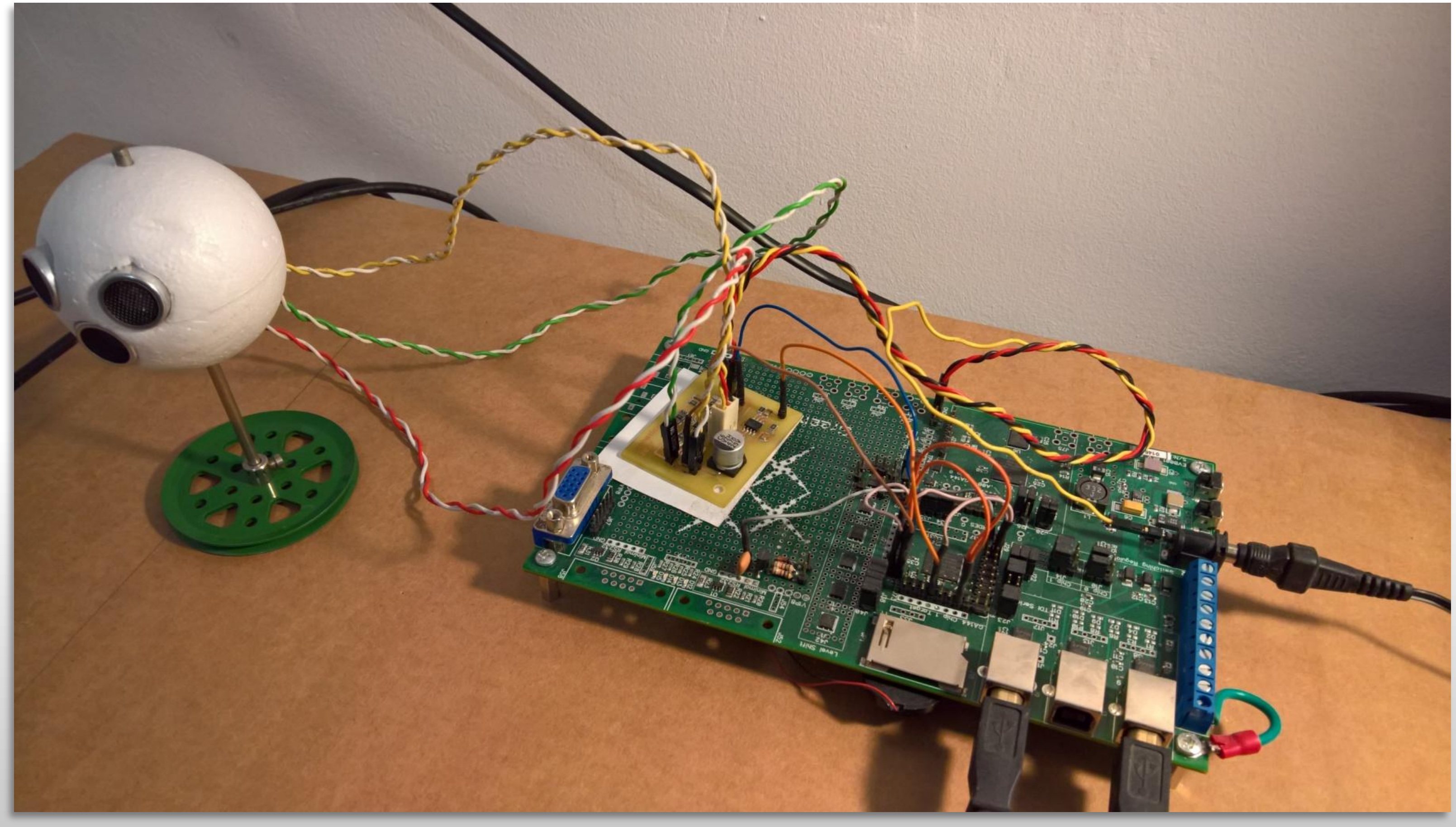


head with ultrasonic transducers



colorForth application

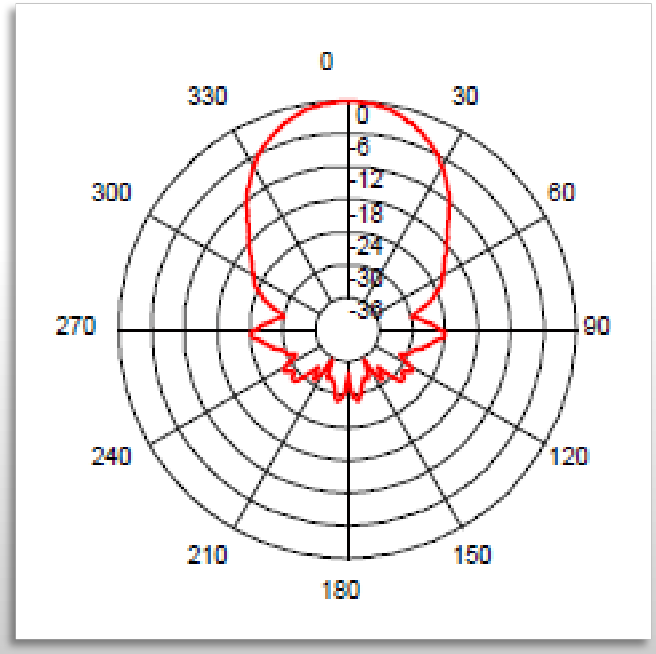
setup



ultrasonic ceramic transducers

Prowave 400ST/R160

central frequency	40.0 kHz
bandwidth (-6 dB)	2.0 kHz
max. drive voltage	20V rms
total beam (-6 dB)	55°



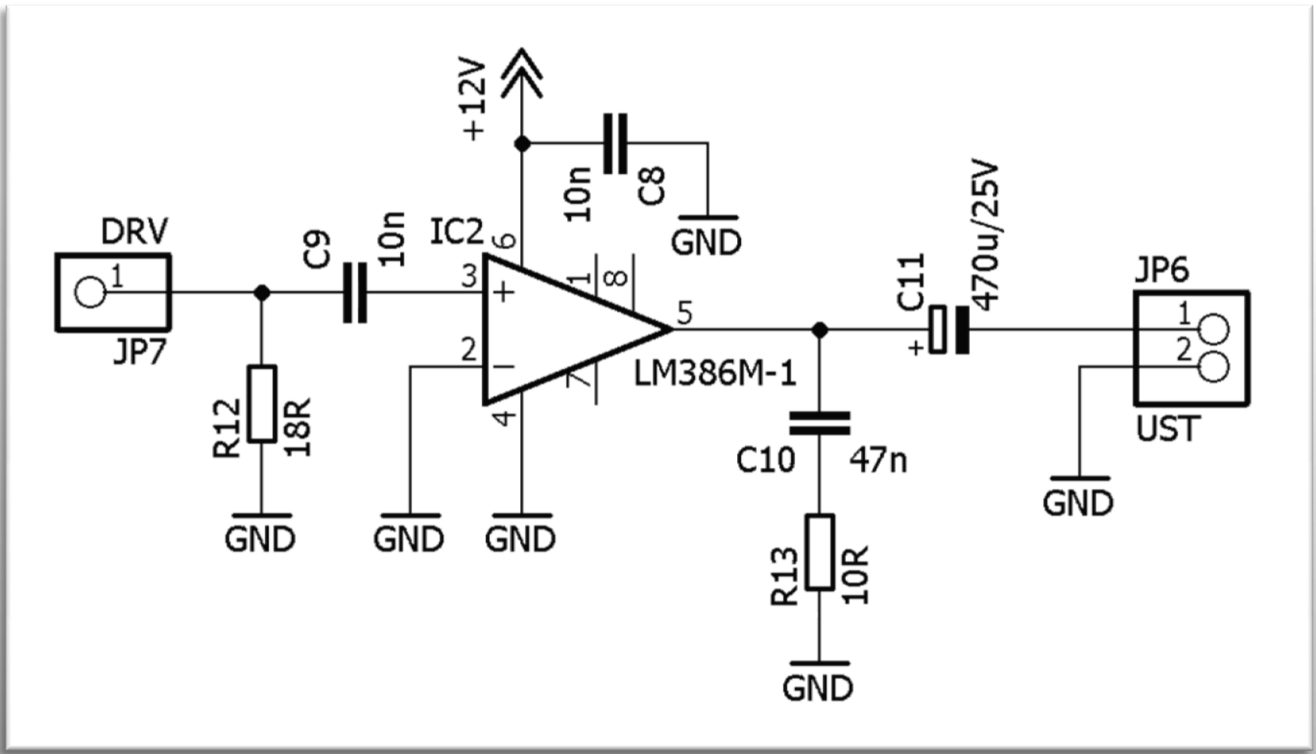
beam angle

Air Ultrasonic Ceramic Transducers 400ST/R160, Prowave datasheet

analog module

transmitter amplifier

input voltage (Vpp)	450 mV
gain	26 dB

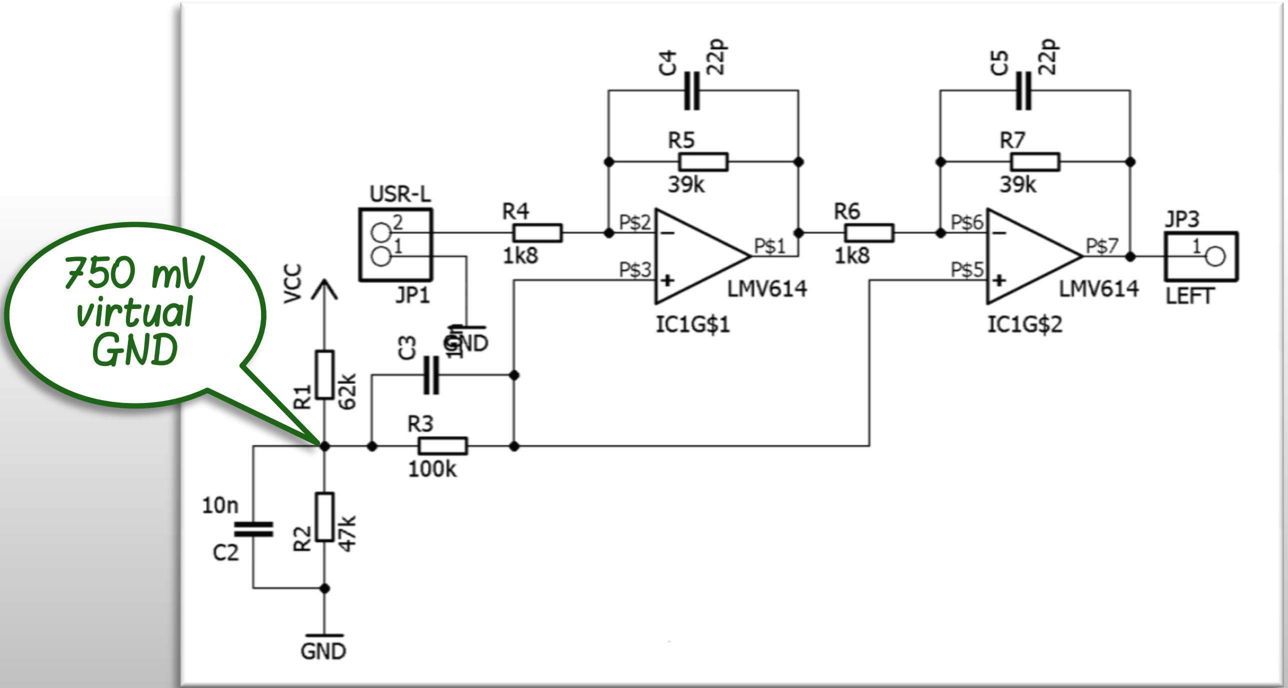


based on LM386 datasheet

analog module

receiver amplifiers

supply voltage	1.8 V
gain	53 dB



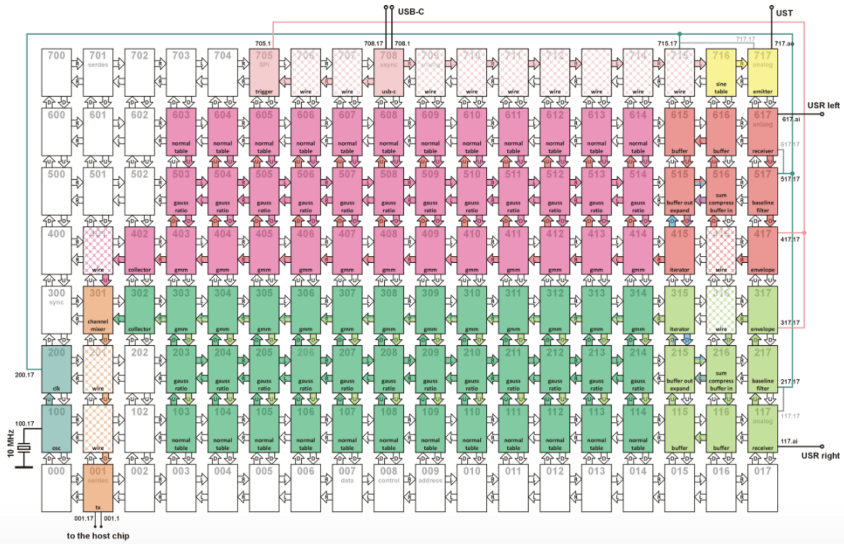
based on TI Application Report SLAA136A, October 2001

FLOORPLAN

floorplan

159 nodes

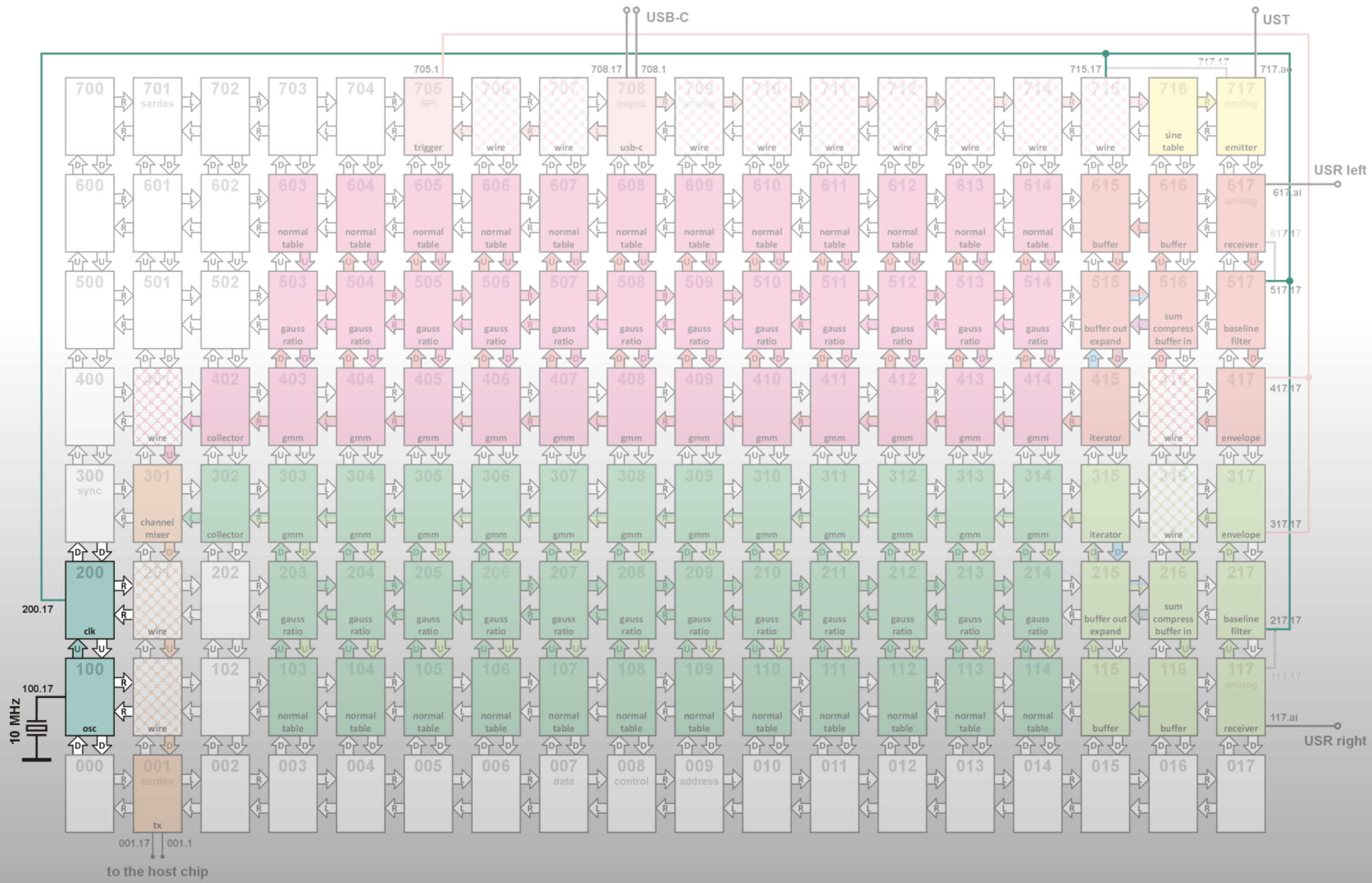
target chip



host chip

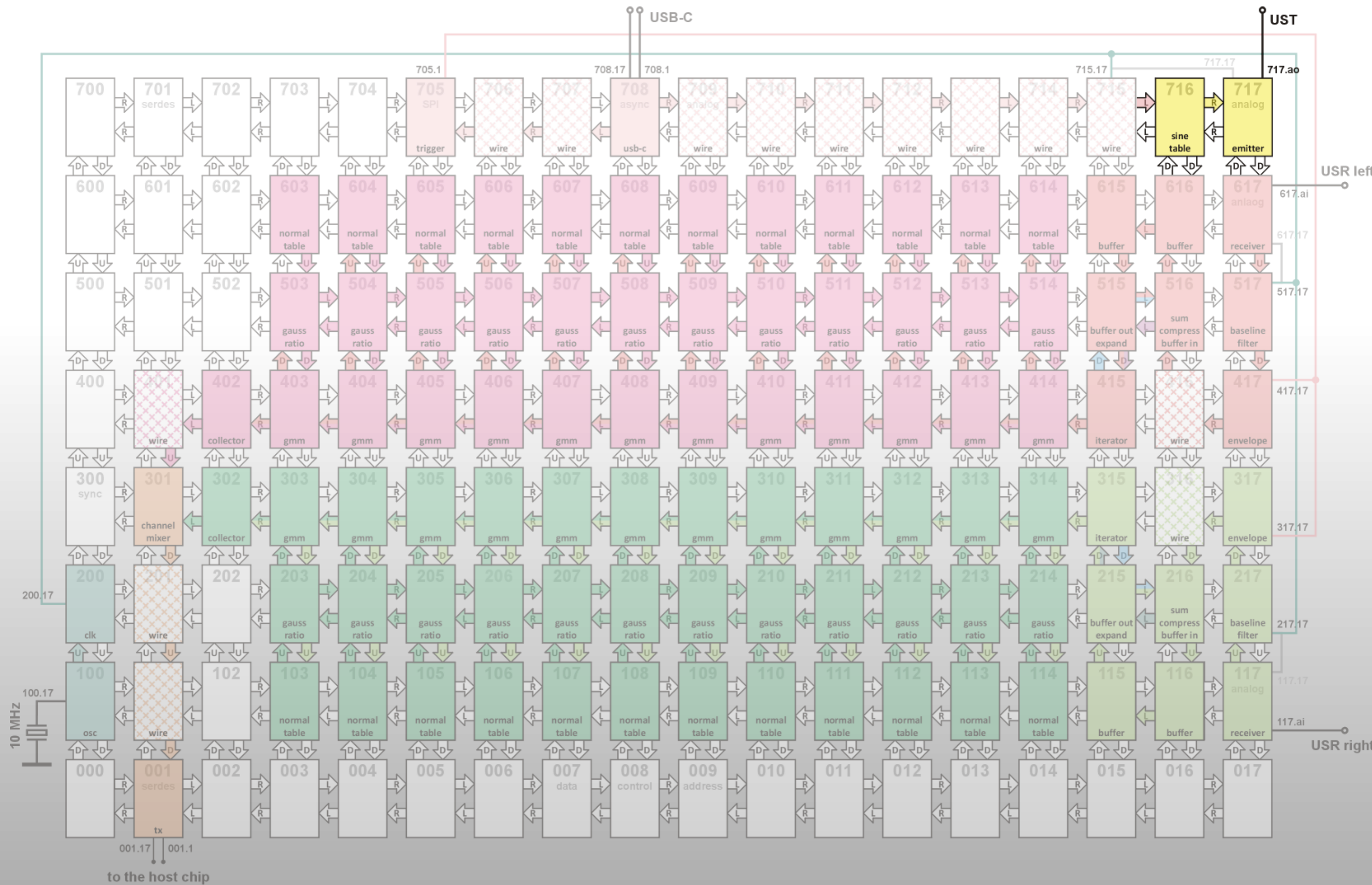
target chip

1 MHz clock



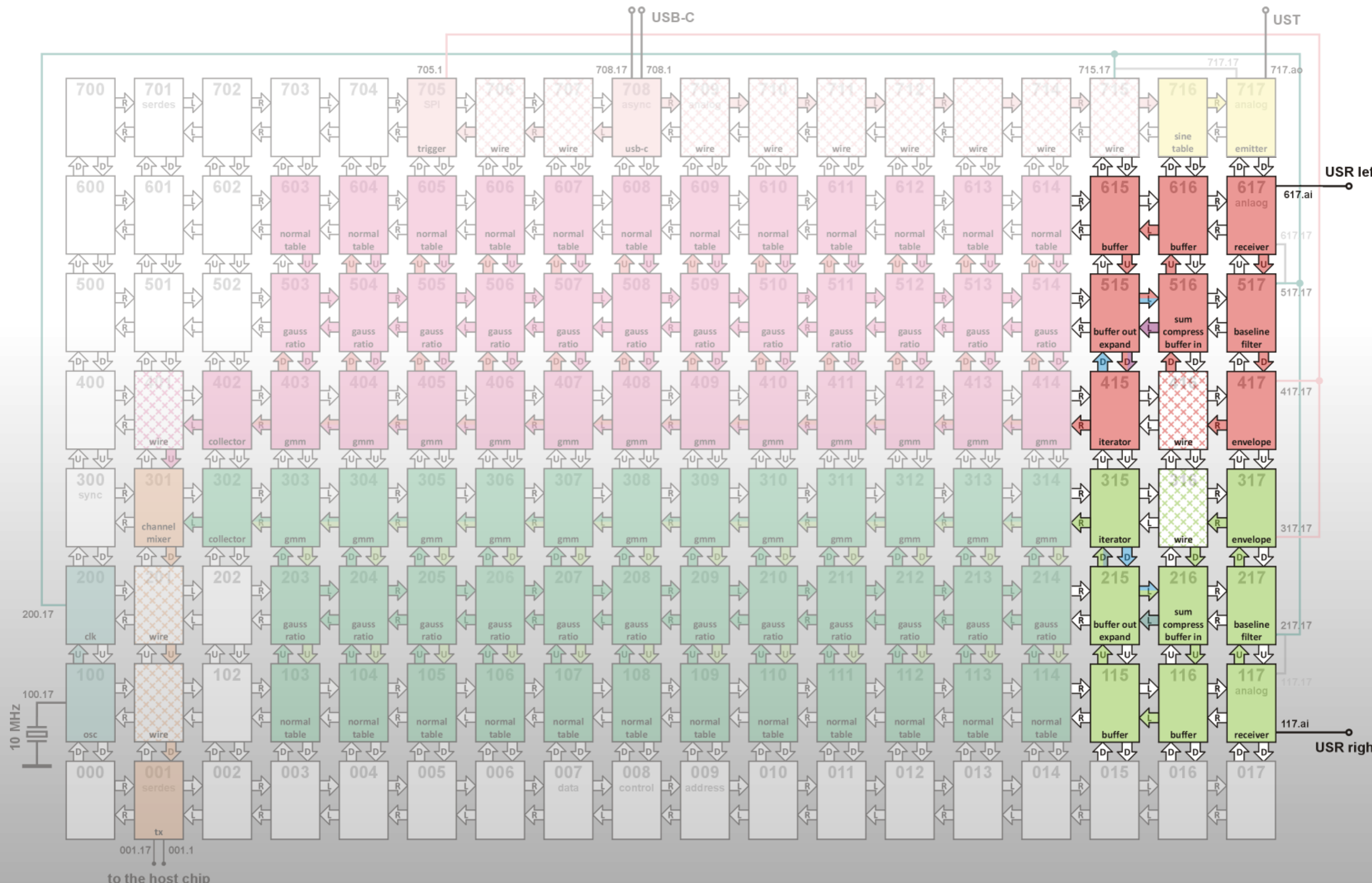
target chip

ping generator



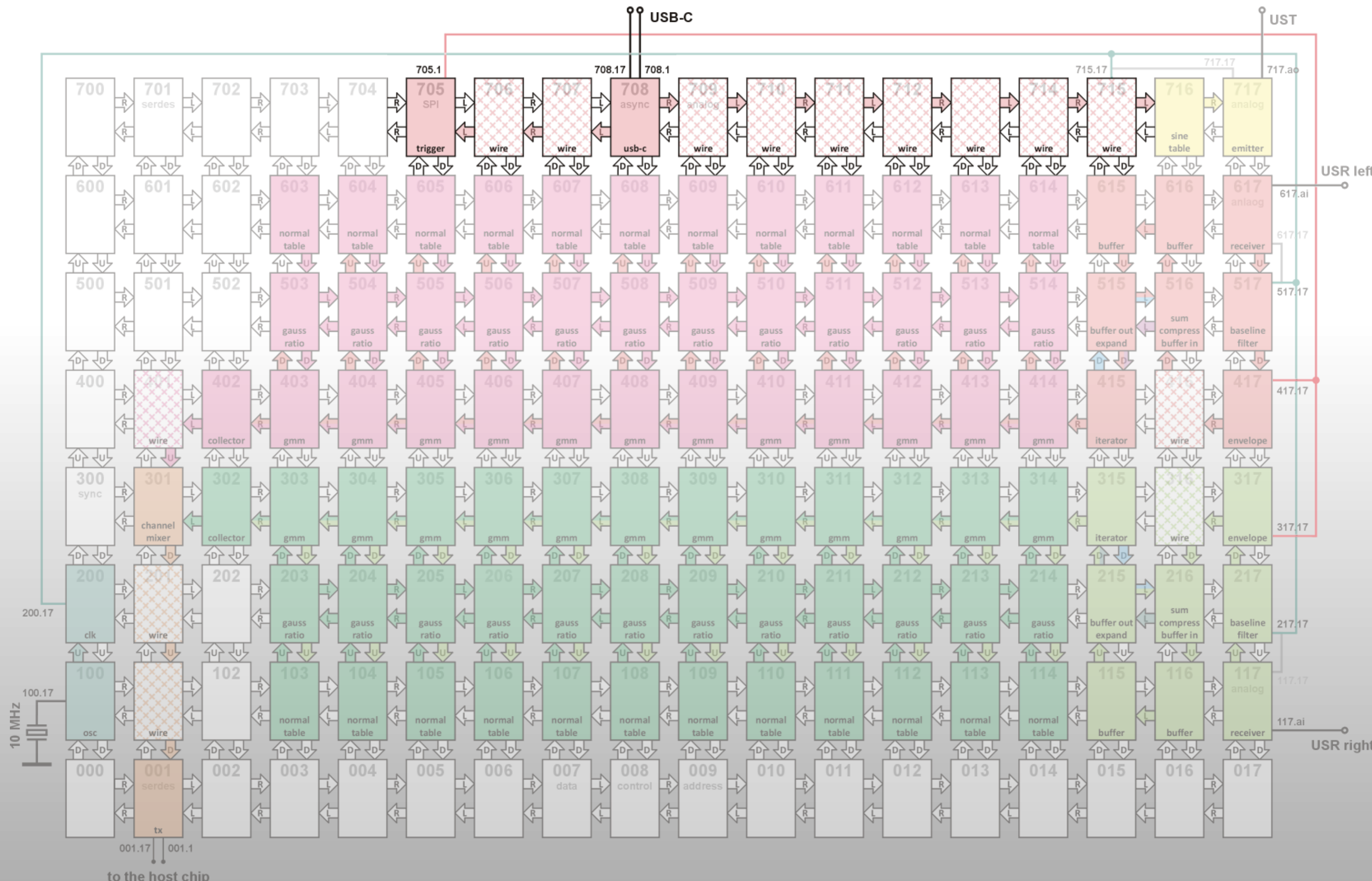
target chip

echo recording modules



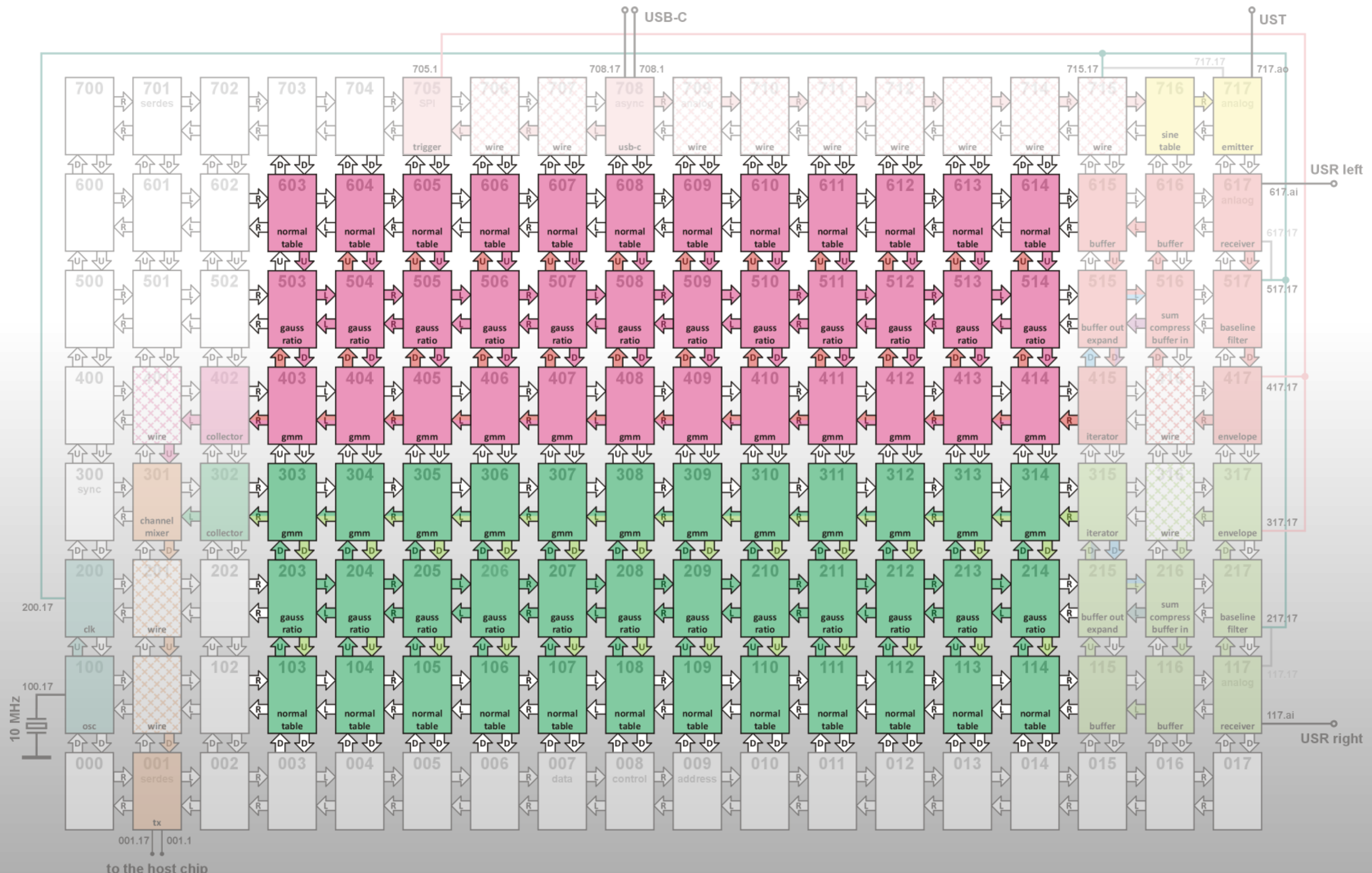
target chip

application trigger



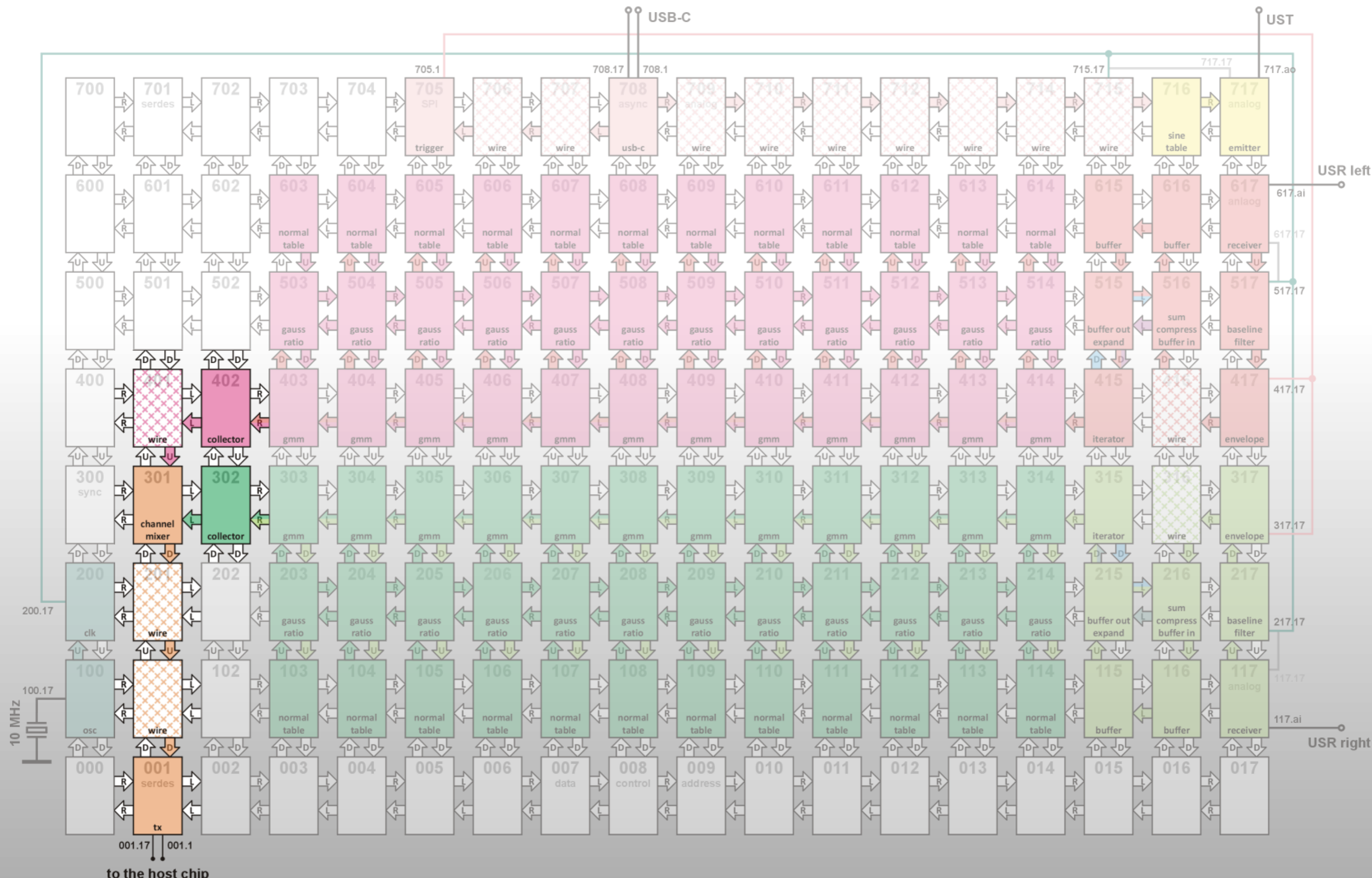
target chip

echo processing modules



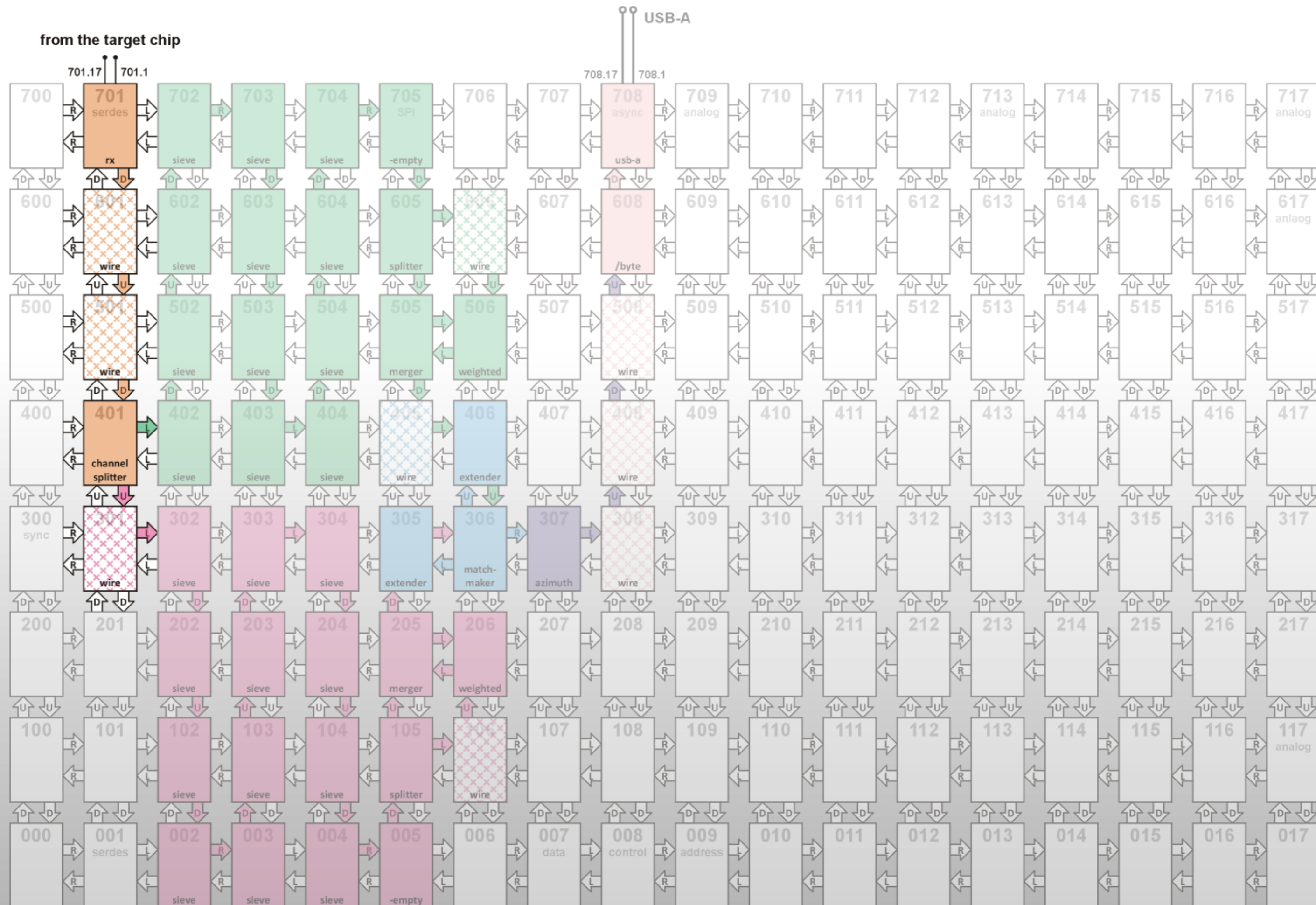
target chip

collectors, channel mixer, serdes



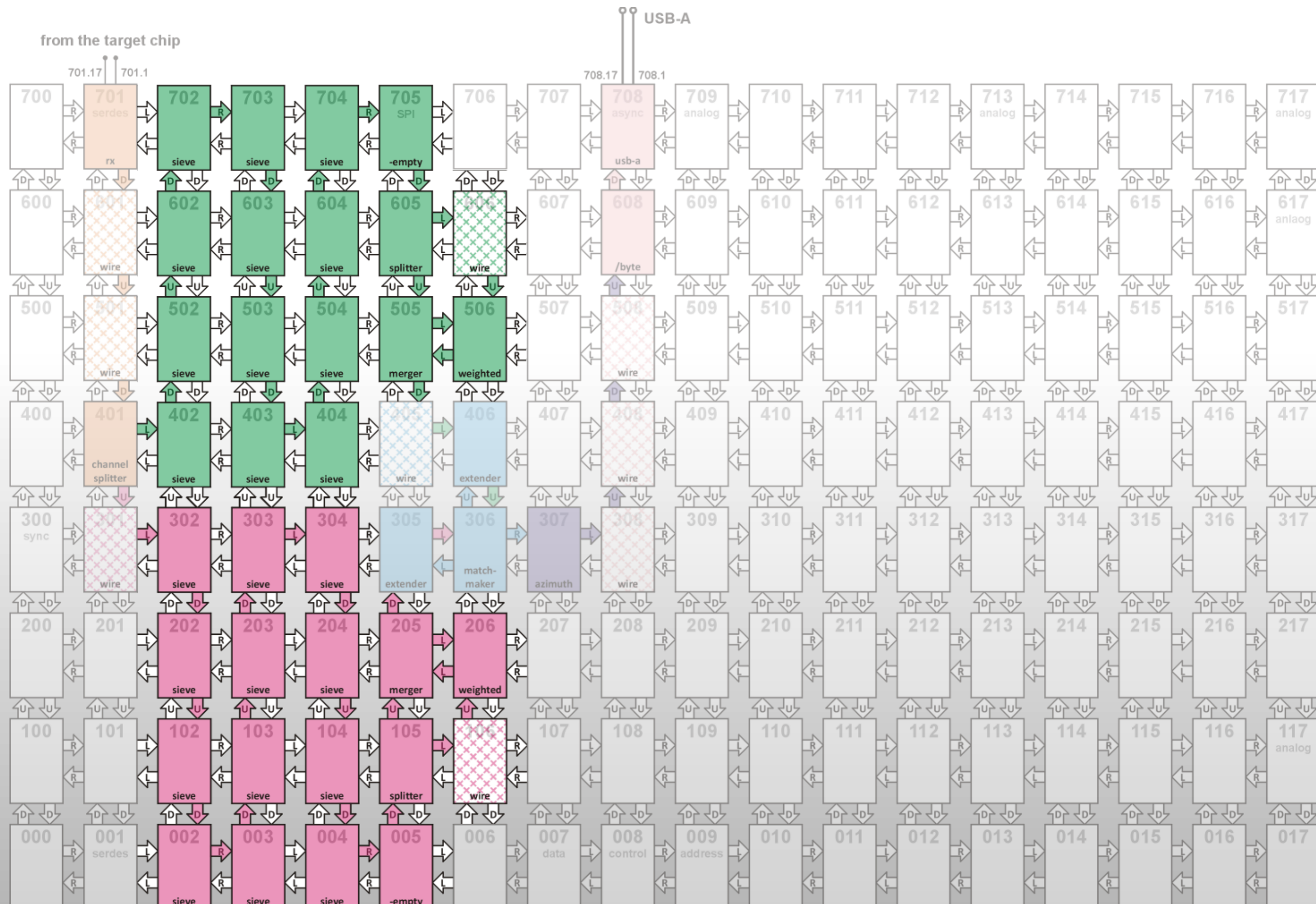
host chip

serdes, channel splitter



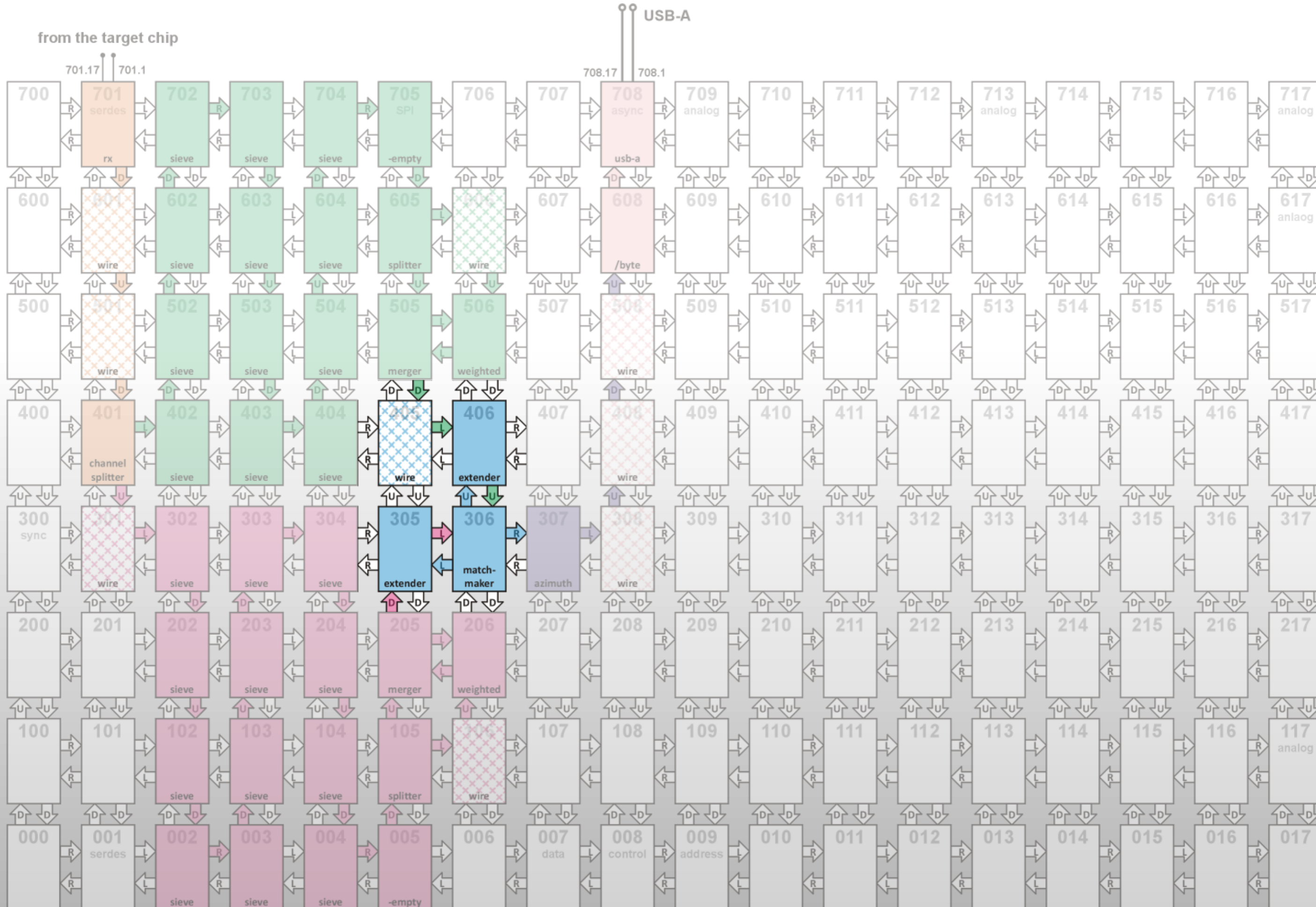
host chip

post-processing modules



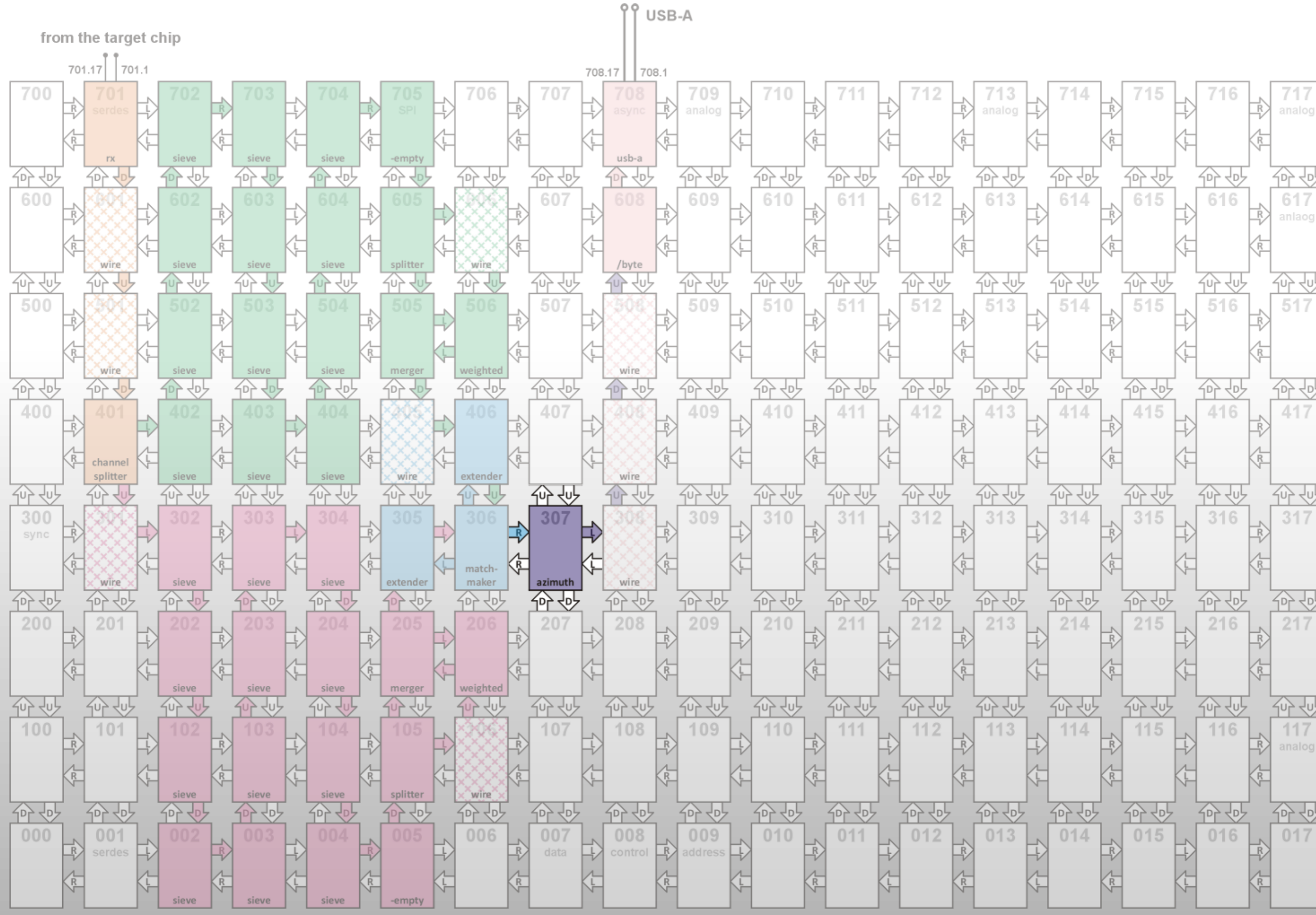
host chip

matchmaker module



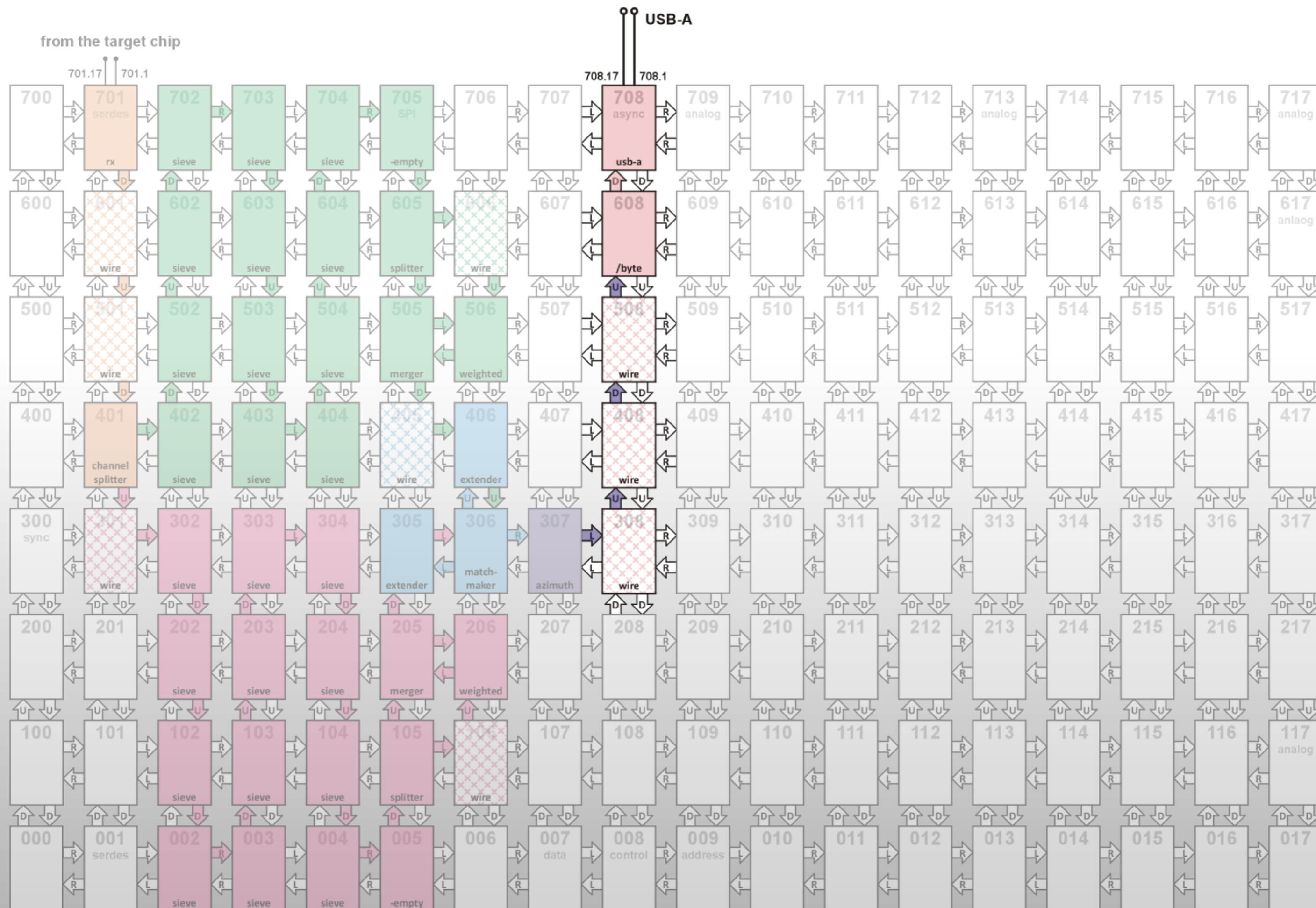
host chip

azimuth calculator



host chip

transfer to PC

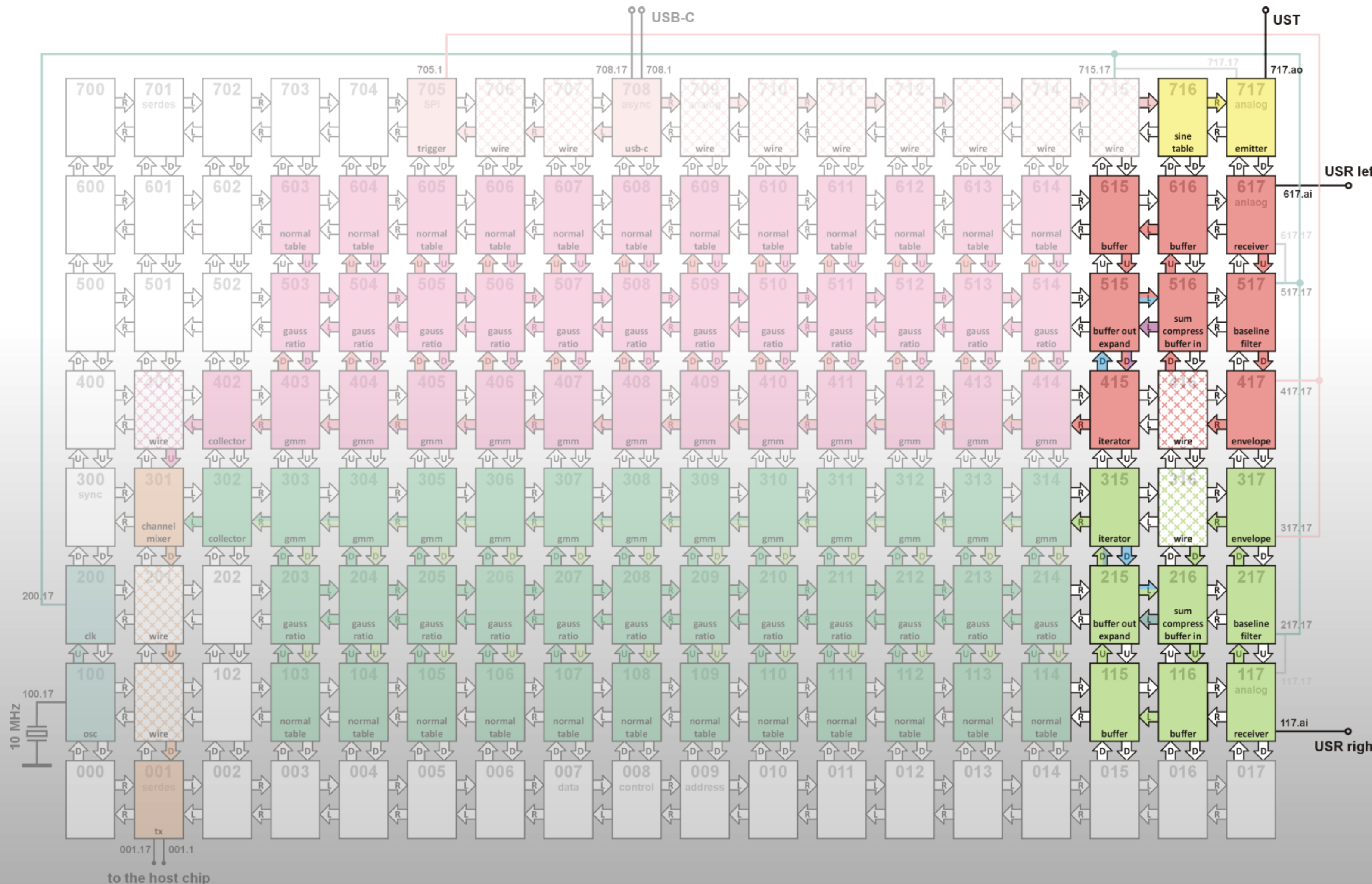


IMPLEMENTATION

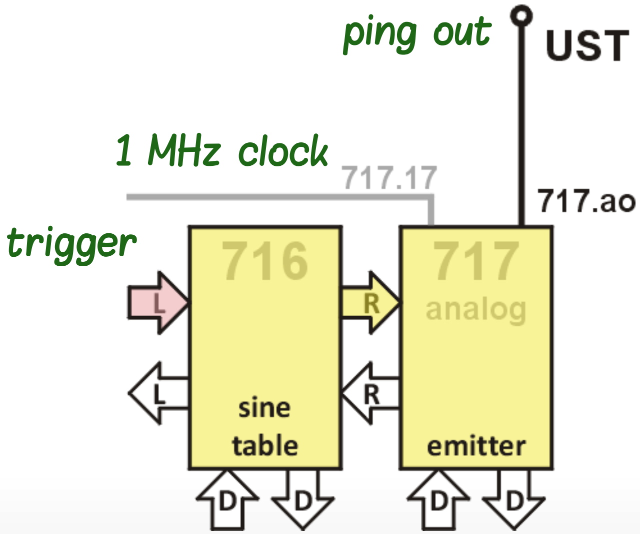
part I
echo recording

target chip

ping generator & echo recording



ping generator



40 kHz sine burst

based on clocked DAC

suitable for:

- synchronous motors (1 kHz)
- audio signal (10 kHz)
- RF signal (100 kHz)
- VGA color sync (25 MHz)

how DAC works?

current source – load resistor to GND

9-bit value

xor with $155

store in IO_{0-8} – immediate conversion

suspend on write to UP according to WD bit:

- WD = 0   wait while shared pin 17 is low
- WD = 1   wait while shared pin 17 is high

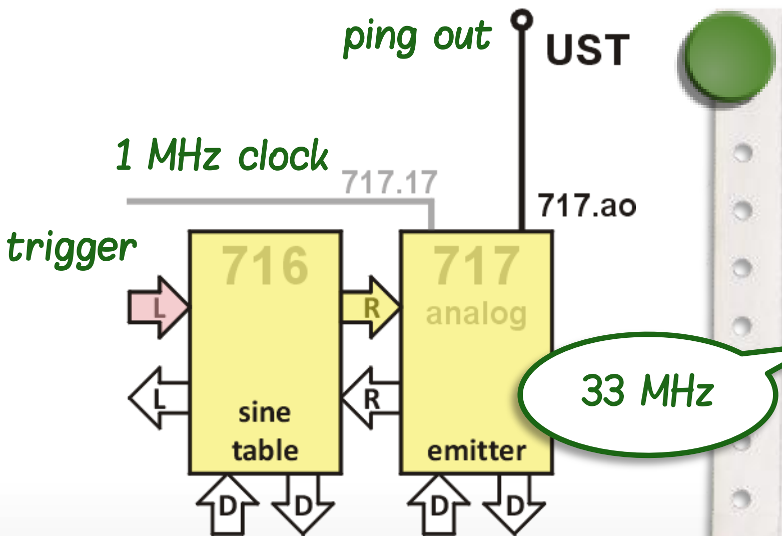
IO and UP not directly addressable

IO in B register, UP in A register

there's a problem: how to read data?

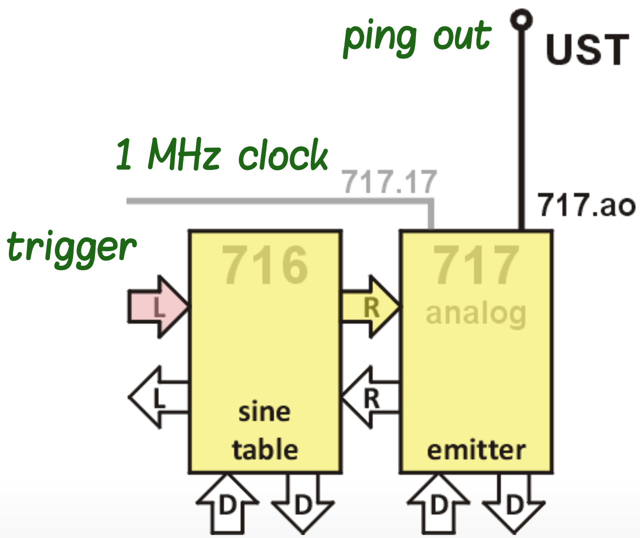
using P register!

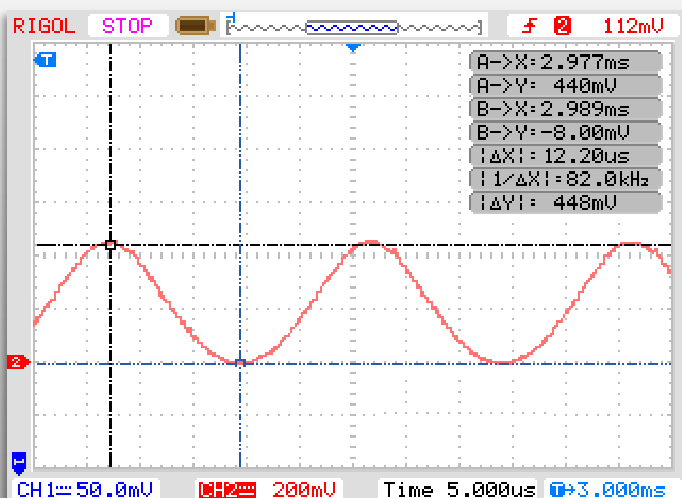
ping generator




```
10717 +node 10717 /ram up /a io /b right /p
15555 15D55 15555 15D55 15555 15D55
15555 15D55 15555 15D55 10 /stack

reclaim 10717 node 0 org
dac 00 n or dup ! !b ;

10716 +node 10716 /ram right /b left /p
5600 5600 5600 5600 5600 5600 5600 5600
5600 5600 @p dac 10 /stack

reclaim 10716 node 32 org
init or a! ;
ping 33 n for init 49 for !b @+ !b unext
next !b 200 !b ;

0 org
200 , 231 , 263 , 295 , 328 ,
360 , 390 , 419 , 444 , 465 ,
482 , 493 , 499 , 499 , 493 ,
482 , 465 , 444 , 419 , 390 ,
360 , 328 , 295 , 263 , 231 ,
...
```

ping generator





actual scope trace

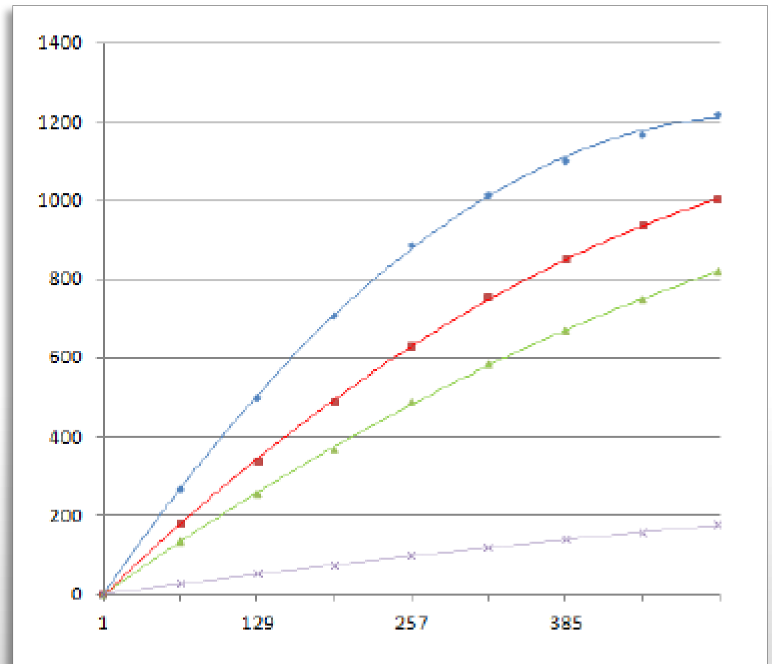


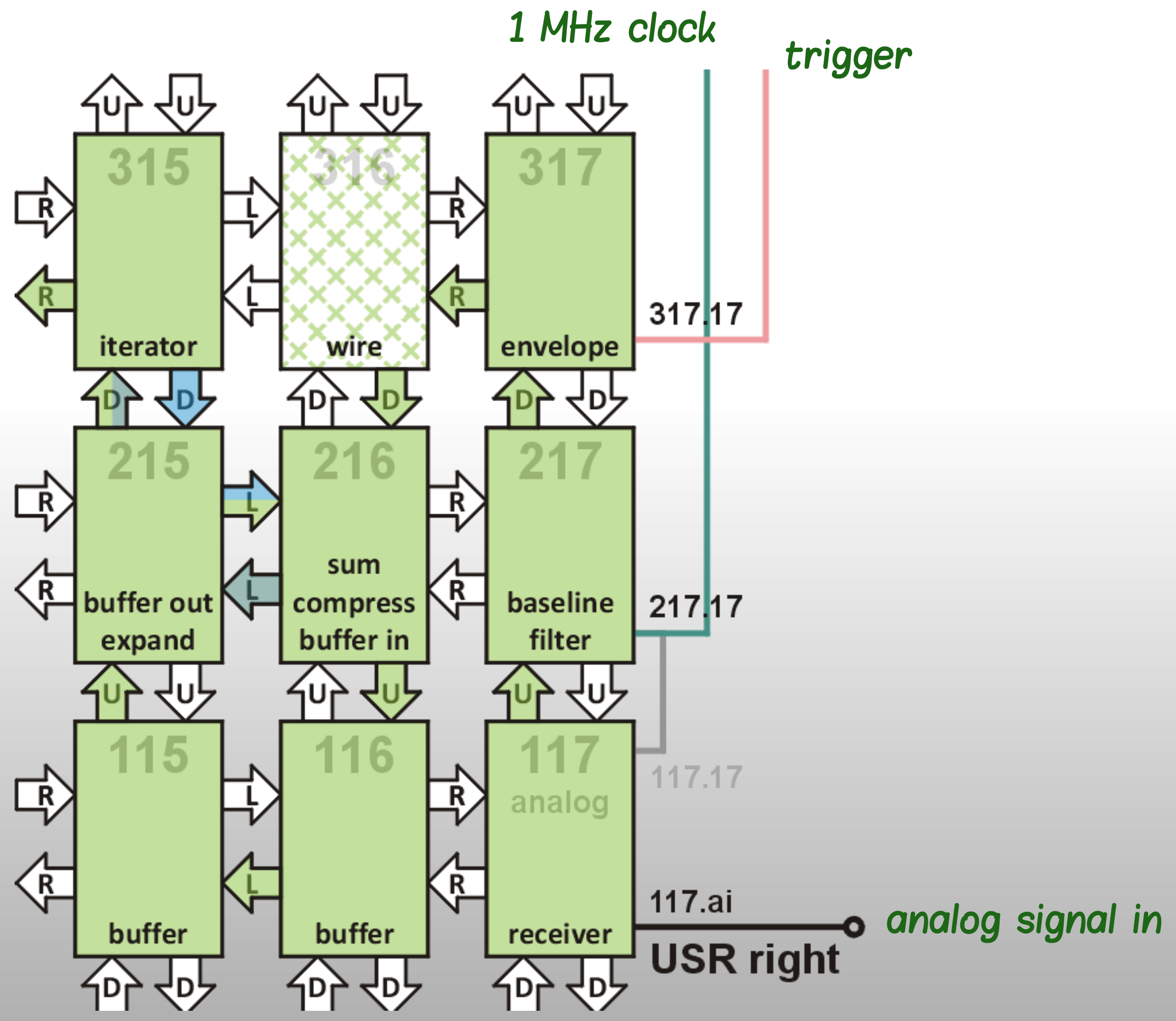
Figure 12 Typical DAC Transfer Function

DB001 F18A Technology Reference

echo recording module

digitizer
buffer
iterator

amplitude envelope out



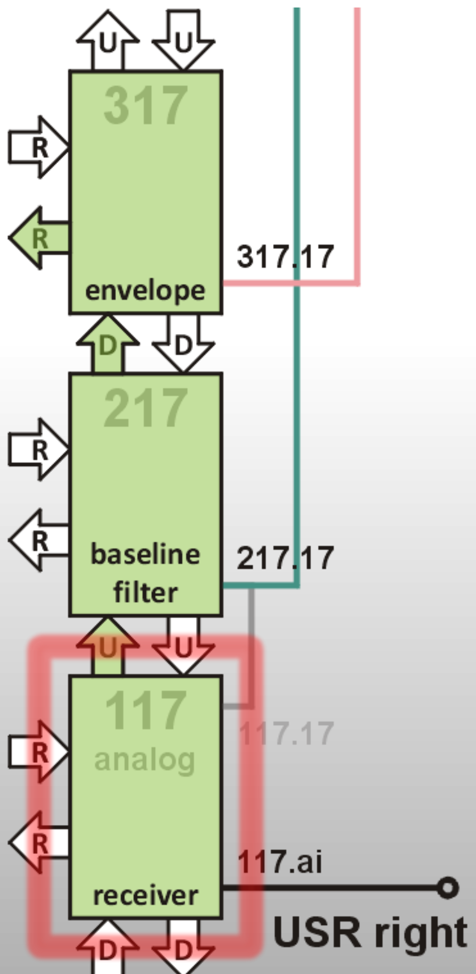
echo recording – digitizer

receiver

VCO from 5.6 GHz (V_{SS}) to 3.6 GHz (V_{DD})

18-bit modulo down-counter

linear between 750 and 1300 mV



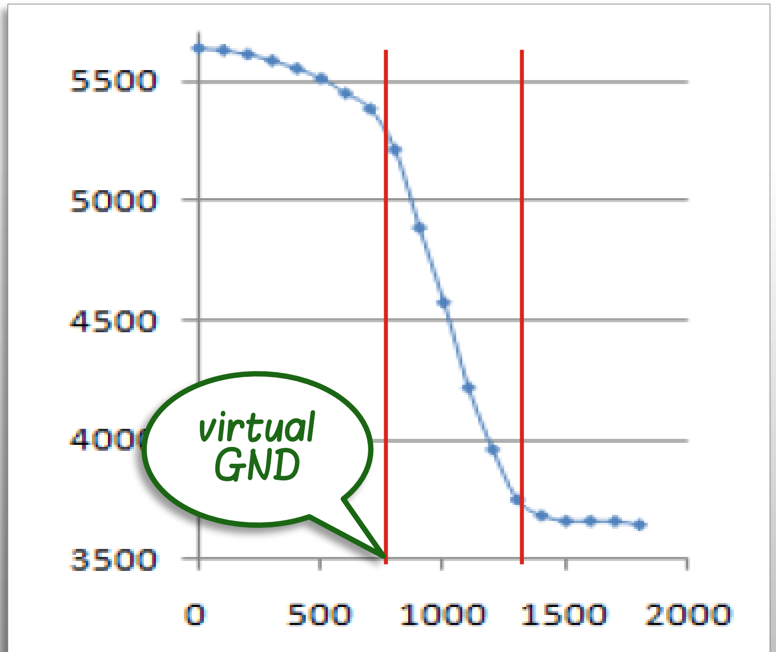
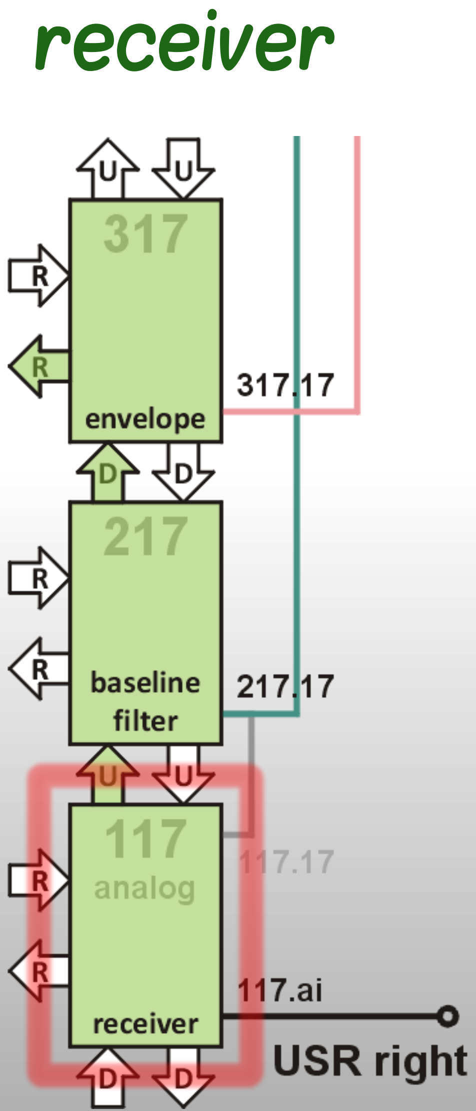


Figure 13 Typical ADC Transfer Function

DB001 F18A Technology Reference

echo recording – digitizer



write to LEFT to stop VCO

- suspended according to WD level

read first count form LEFT

- VCO starts running

wait – voltage sampling interval

write to LDATA to stop VCO

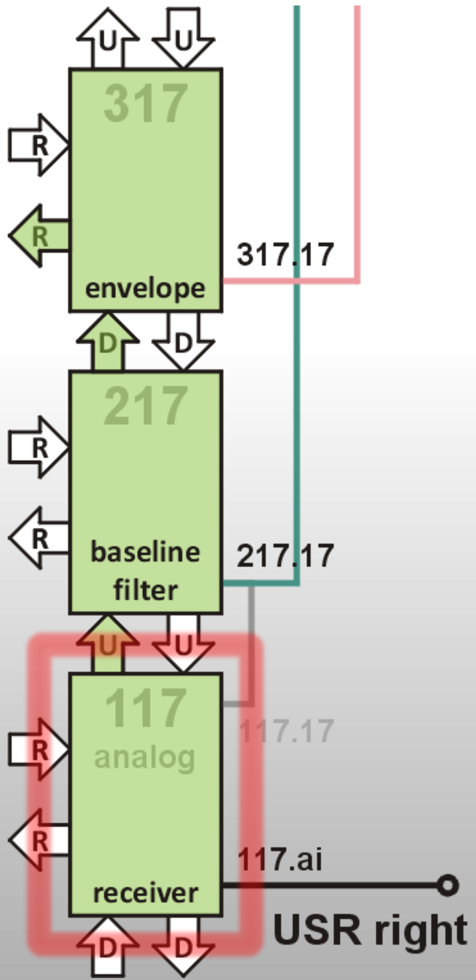
- F18 not suspended

read second count from LDATA

count difference proportional to voltage

echo recording – digitizer

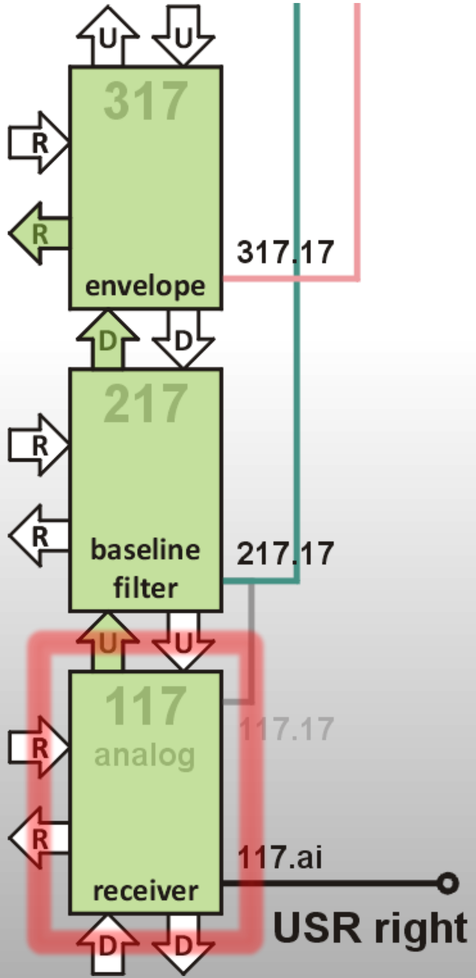
receiver

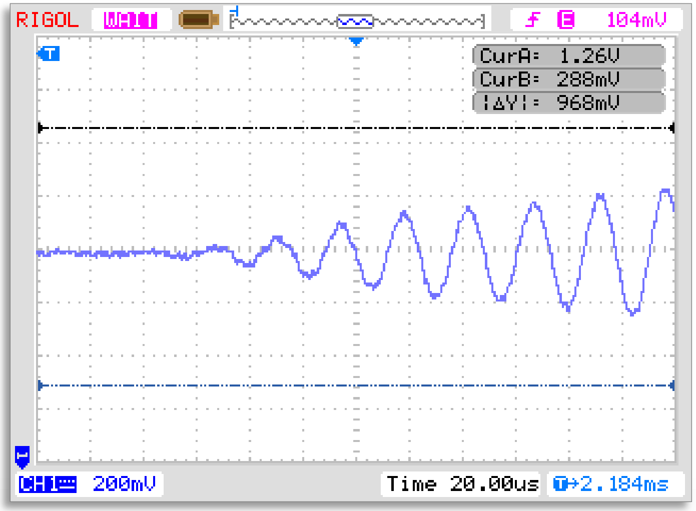



```
10117 +node 10117 /ram io /b 12 go /p
   11D55 11555 11D55 11555 11D55 11555
   11D55 11555 11D55 11555 10 /stack

reclaim 10117 node 0 org
1+ 1 . + ;
dif - . + -if - ; then 1+ ;
adc left a! !b dup ! @ 171 ldata a!
35 for unext dup ! @ dif ;
send up b! !b io b! ;
go 12 adc -200 . + send go ;
```

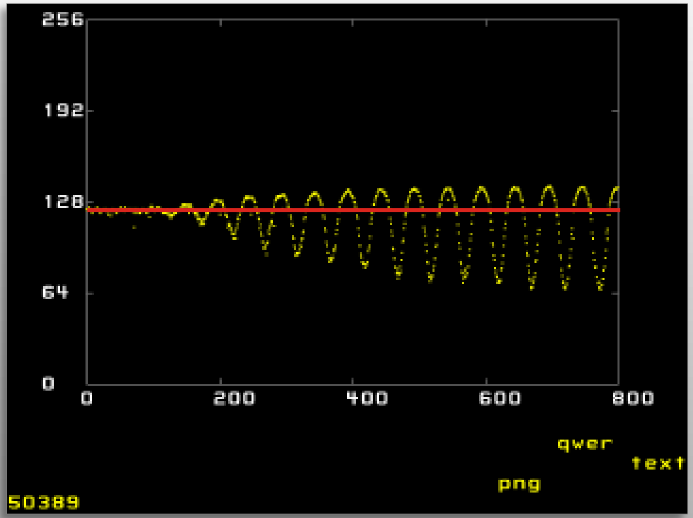
echo recording – digitizer

receiver





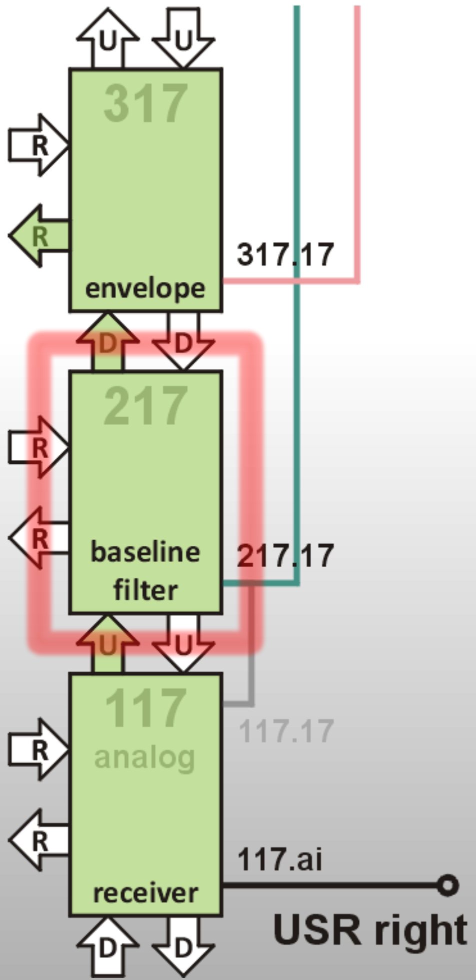
scope trace



ADC output

echo recording – digitizer

filter & baseline



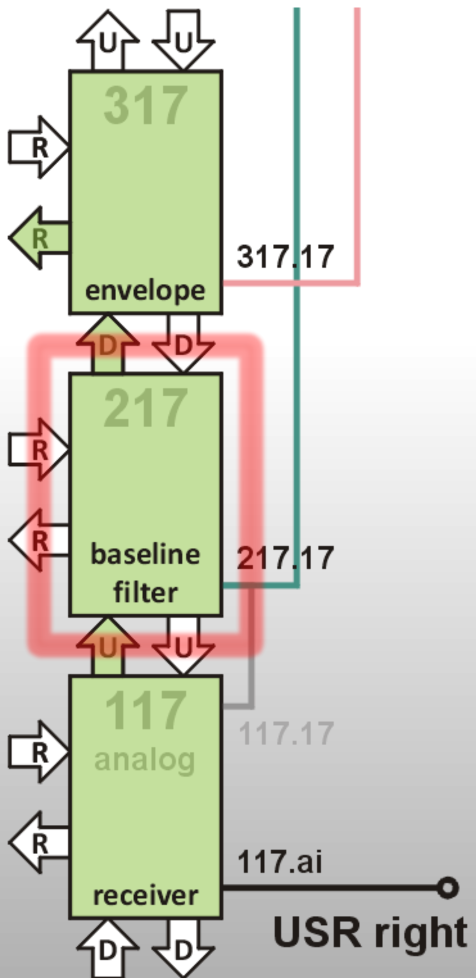
measures baseline

grabs samples from analog node

applies EMA filter

echo recording – digitizer

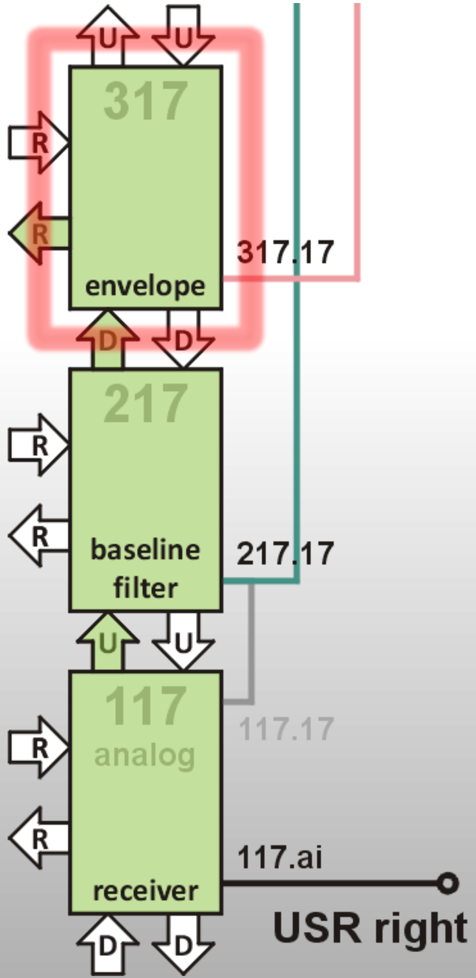
filter & baseline




```
10217 +node 10217 /ram down /a up /b
  19999 0.8 1 /stack 25 go /p

reclaim 10217 node 0 org
1+ 1 . + ;
*.r a! dup dup or
16 for +* unext a -if drop 1+ ; then drop ;
ema *.r over - 1+ 1FFFF and
@b *.r a! drop a . + dup ;
flush @ drop ;
base dup dup or 255 for @b . + next
2/ 2/ 2/ 2/ 2/ 2/ 2/ 2/
dup - 1+ down a! ! ;
grab 16000 for ema down a! ! next - ! ;
go 25 dup ! flush base flush grab go ;
```

echo recording – digitizer

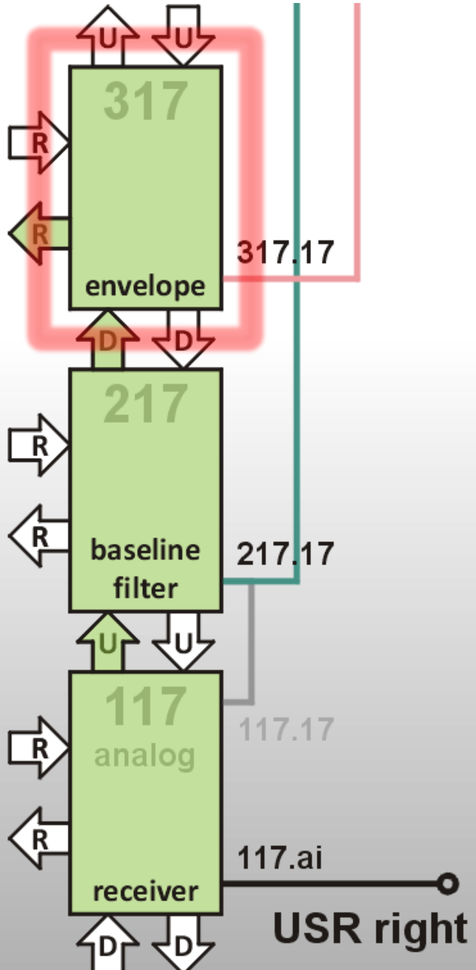
envelope

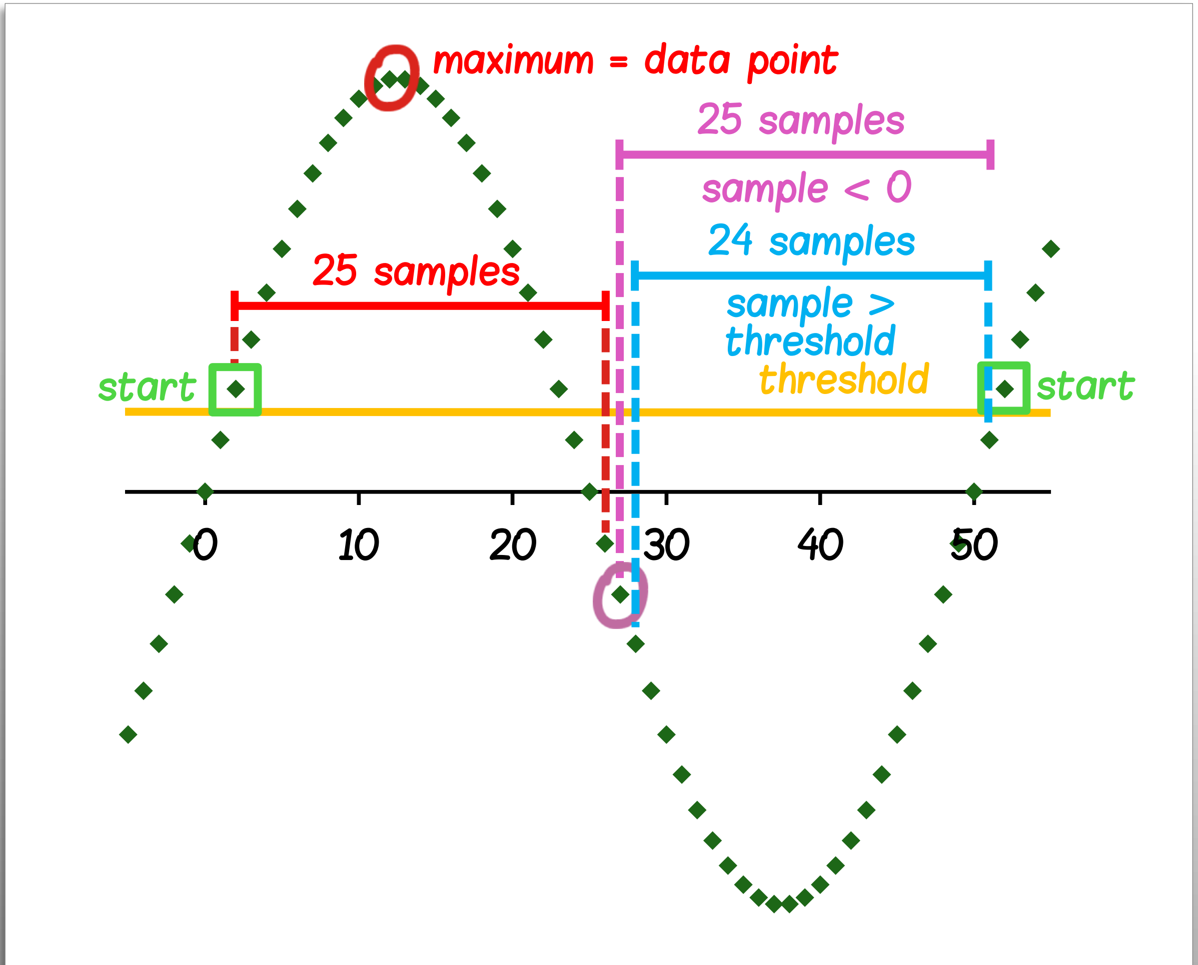


- initiates conversion upon trigger signal
- removes direct sound wave
- detects positive halfwaves
- finds peak maxima
- inverts digitized values
- subsamples data (1:50)
- self synchronizing algorithm

echo recording – digitizer

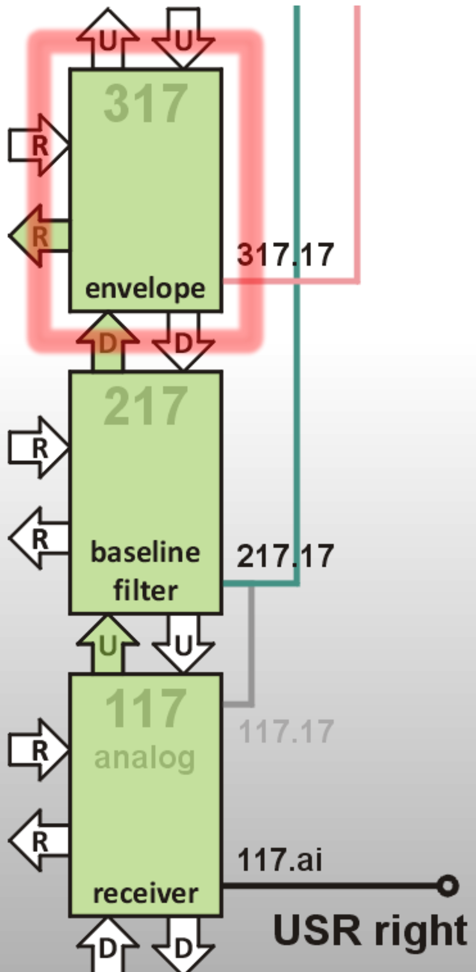
envelope

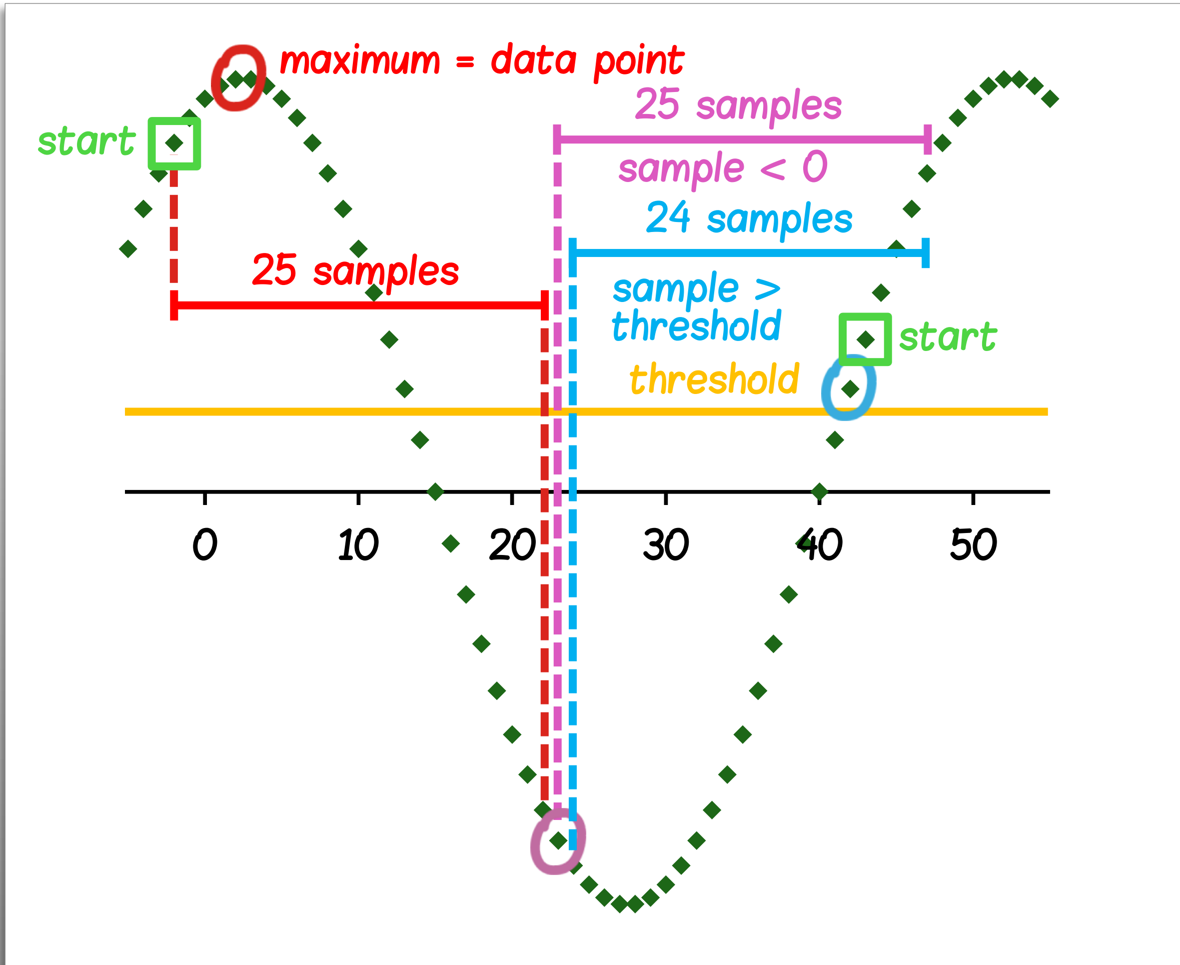




echo recording – digitizer

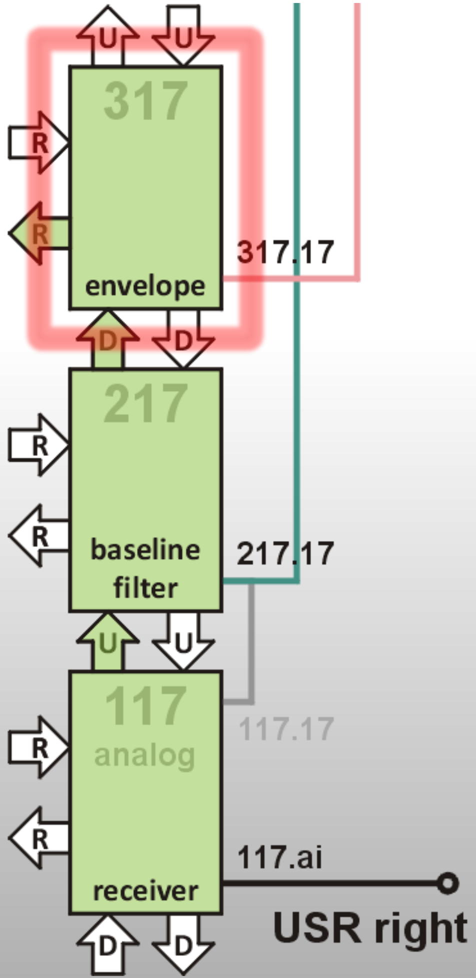
envelope

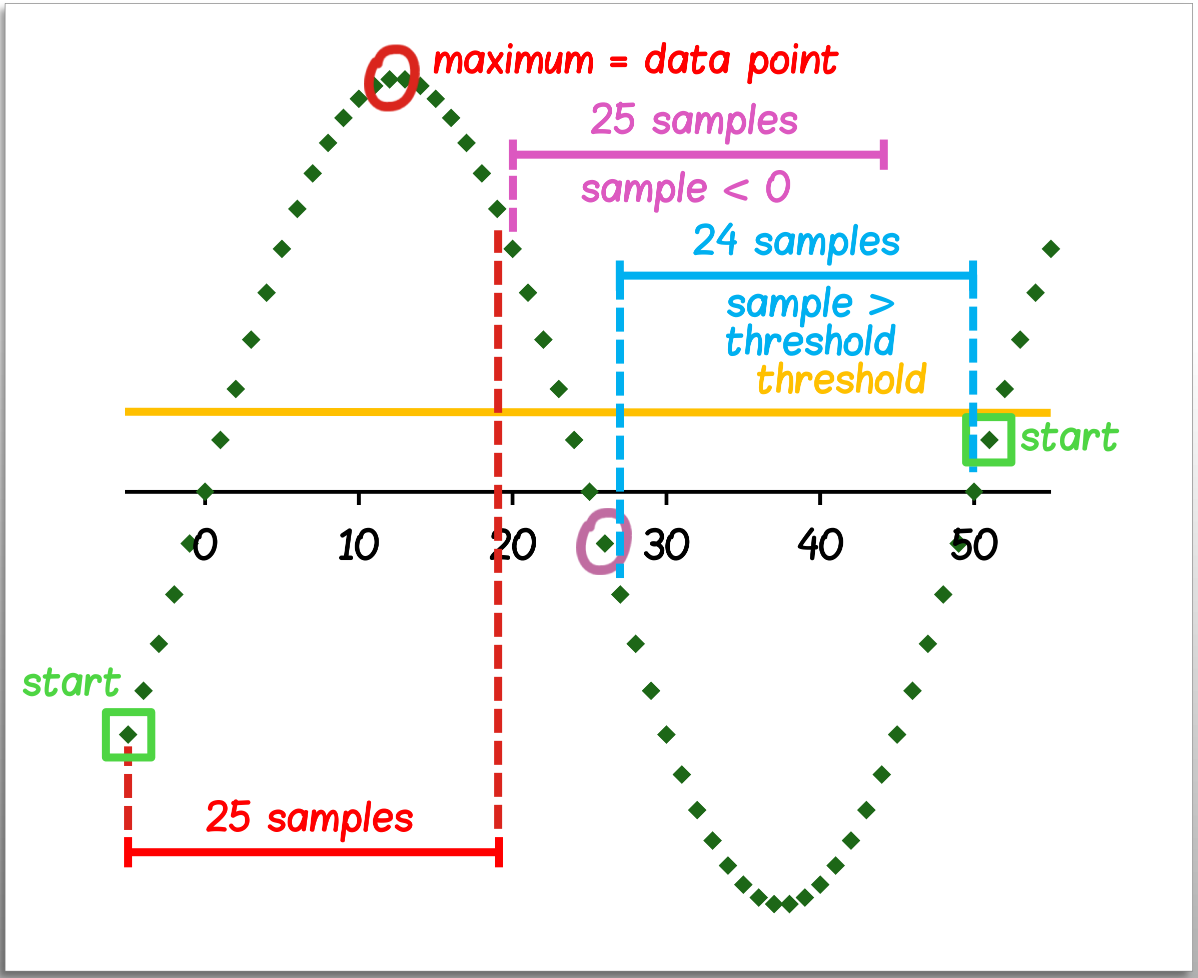




echo recording – digitizer

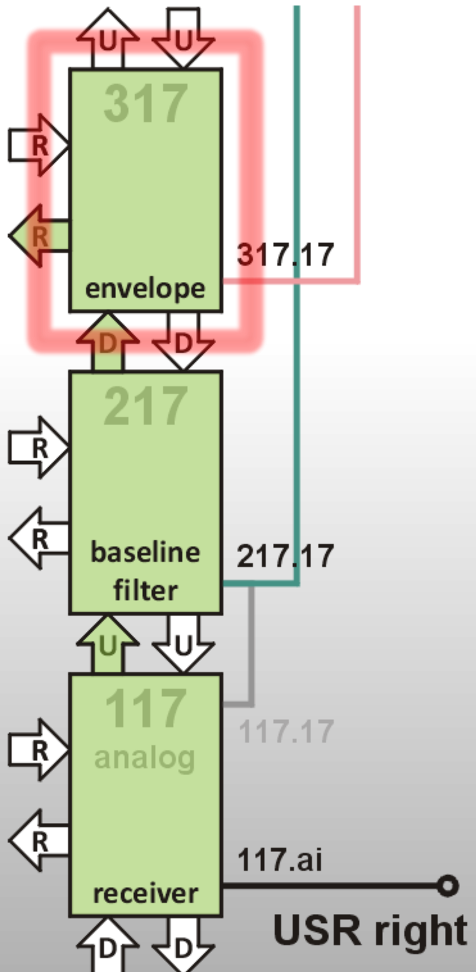
envelope

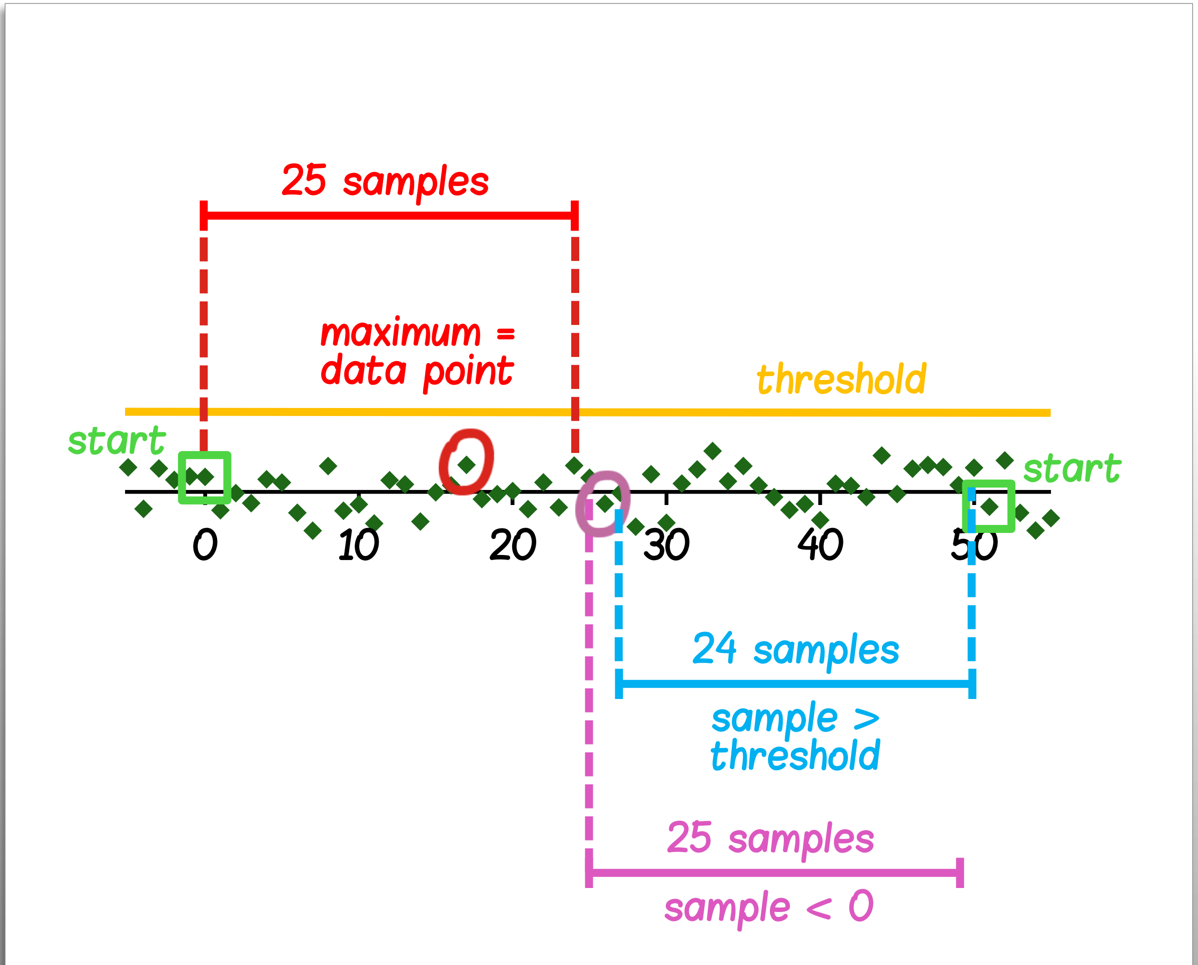




echo recording – digitizer

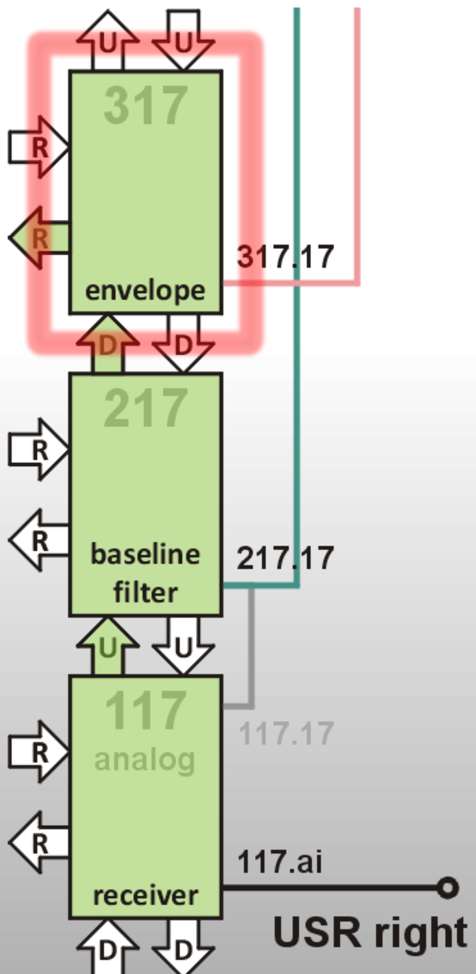
envelope





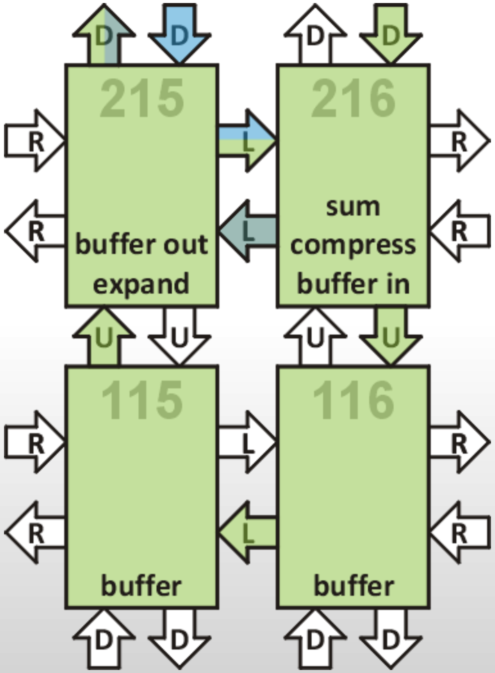
echo recording – digitizer

envelope




```
10317 +node 10317 /ram 195 rd-- /a left /b
30 env /p

reclaim 10317 node 0 org
ap @p drop @p ; ap! !p ; .. 0 ,
1+ 1 . + ;
x n-nx @ over . + ;
min - over . + - -if + ; then drop ;
trig n dup begin drop x -until drop drop ;
apex -n 0 ap! 24 for x ap min ap! next ap ;
fin dup 24 for x -if drop swap next ;
then drop pop drop 23 push
8 . + begin x - -if drop swap next drop ;
then drop drop pop drop ;
flush begin @ -until ;
inv/ n-m - 1+ -if dup or then ;
-ping 499 for @ drop unext ;
env 30 @b @ @ -ping dup 8 . + trig -ping
255 for apex inv/ ! fin next flush env ;
```

echo recording – buffer



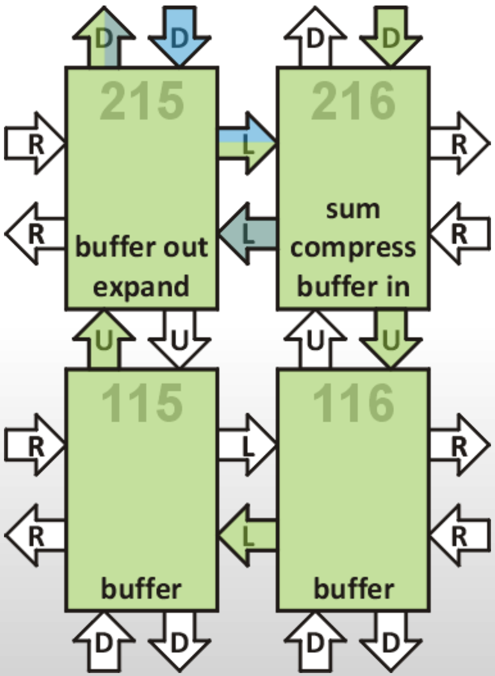
256 bytes packed in 128 18-bit words

calculates sum of data points

circular buffer controlled by iterator

based on Greg Bailey's delay line (see AB005)

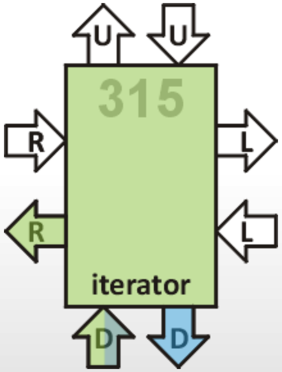
echo recording – buffer




```
+dly ion +node /b /p FBB2 9E27
over over over over over over over over
10 /stack 0 /a ;
up left 10116 +dly left up 10115 +dly

10216 +node 10216 /ram 135 -dl- /a up /b ...

reclaim 10216 node 0 org
...
shove n @p !b .. !b @ !b @p
!b @p !b .. !+ !b .. ;
...
```

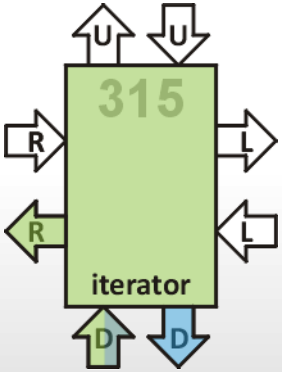
based on Greg Bailey's delay line (see AB005)

echo recording – iterator



controls circular buffer (0/-1 flags)
cycles through iterations (10 times)
polls right port – request for new data

echo recording – iterator




```
10315 +node 10315 /ram down /a 16 go /p

reclaim 10315 node 0 org
more? -f io b! begin @b - 10000 and until ;
wait 50 for . . unext ;
round 255 for more? ! right b! @ !b next ;
go 16 9 for 0 ! wait round wait next
-1 ! go ;
```

DEMO #1

echo recording

[link to demo #1 video on YouTube]

IMPLEMENTATION

part II
echo processing

target chip

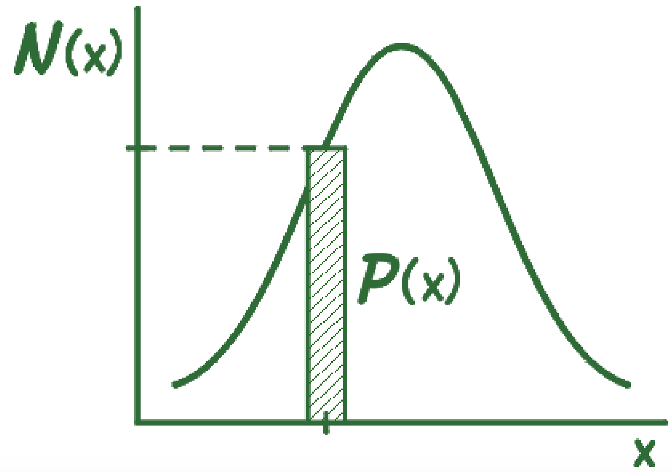
echo processing

math concepts

normal distribution

probability density function (pdf)

- mean μ
- variance σ^2



$$N(x\,|\,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

discrete and integer data $P(x) = N(x)$

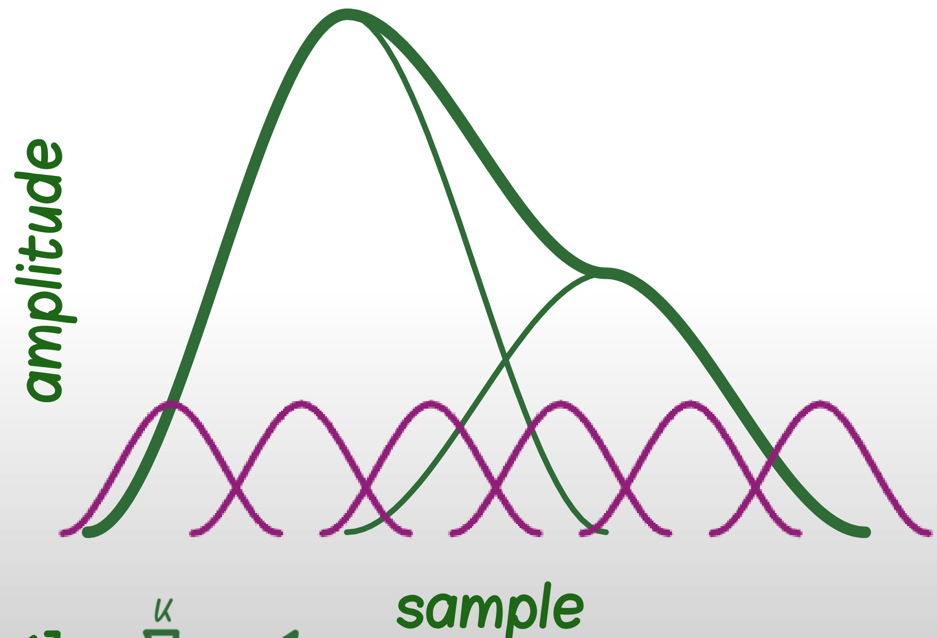
math concepts

gaussian mixture model (gmm)

sampled signal

- sample x
- amplitude y

composed of K gaussians



$$y = \sum_{j=1}^{K} \alpha_j N(x \mid \mu_j, \sigma_j^2)$$

mixing coefficient $\alpha_j = [0, 1]$ $\sum_{j=1}^{K} \alpha_j = 1$

$K, \alpha_j, \mu_j, \sigma_j^2$ hidden variables collectively denoted Θ

math concepts

expectation – maximization (EM) algorithm

finds the maximum likelihood Θ

- E-step: calculate the posterior probability $P(\Theta|x)$ using Bayes' theorem
- M-step: update Θ based on the posterior probability $P(\Theta|x)$

math concepts

expectation – maximization (EM) algorithm

finds the maximum likelihood Θ

- E-step: calculate the posterior probability $P(\Theta|x)$ using Bayes' theorem
- M-step: update Θ based on the posterior probability $P(\Theta|x)$

likelihood function

prior probability

$$P(\Theta|x) = \frac{P(x|\Theta)\,P(\Theta)}{P(x)}$$

posterior probability

normalization constant

EM calculation

too complex for GA144

data normalization

$$y_i = \frac{y_i^o}{\sum_{i=1}^{N} y_i^o} \qquad \sum_{i=1}^{N} y_i = 1$$

E-step: posterior probability $P(\Theta|x)$ = responsibility r_j

$$r_j = \frac{\alpha_j N(x\,|\,\mu_j, \sigma_j^2)}{\sum_{k=1}^{K} \alpha_k N(x\,|\,\mu_k, \sigma_k^2)} \qquad r_j = [0, 1]$$

M-step: update hidden variables

$$\hat{\alpha}_j = \sum_{i=1}^{N} r_j\, y_i \qquad \hat{\mu}_j = \frac{\sum_{i=1}^{N} r_j\, y_i\, x_i}{\sum_{i=1}^{N} r_j\, y_i} \qquad \hat{\sigma}_j^{\,2} = \frac{\sum_{i=1}^{N} r_j\, y_i\, [(x_i - \hat{\mu}_j)^2 + \frac{1}{12}]}{\sum_{i=1}^{N} r_j\, y_i}$$

M. N. Stuttle, PhD thesis, University of Cambridge, 2003

EM calculation – simplified

E-step: posterior probability $P(\Theta|x)$ = responsibility r_j

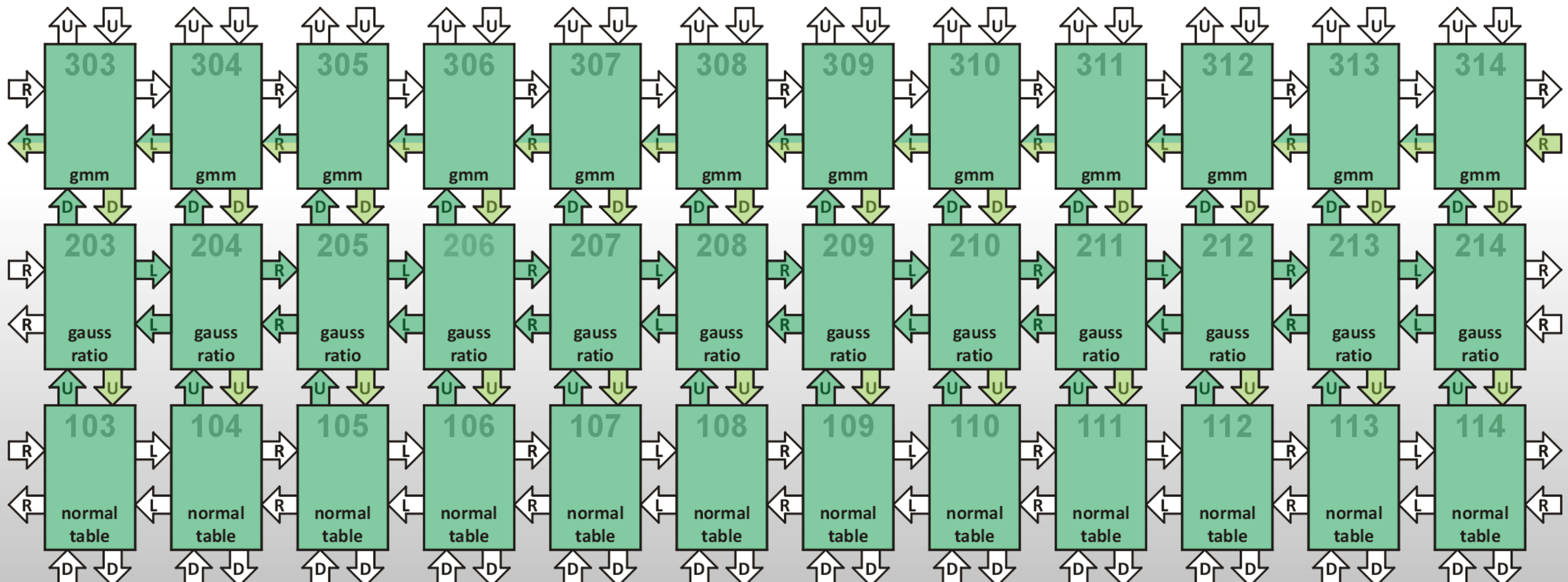
$$r_j = \frac{\alpha_j N(x \mid \mu_j, \sigma^2_{fixed})}{\sum_{k=1}^{K} \alpha_k N(x \mid \mu_k, \sigma^2_{fixed})} \qquad r_j = [0, 1]$$

M-step: update hidden variables

$$\hat{\alpha}_j = \sum_{i=1}^{N} r_j\, y_i \qquad \hat{\mu}_j = \frac{\sum_{i=1}^{N} r_j\, y_i\, x_i}{\sum_{i=1}^{N} r_j\, y_i}$$

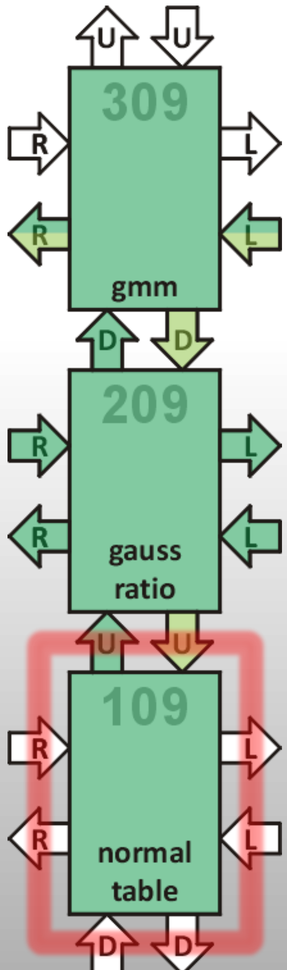
echo processing module

implementing gaussian mixture model



gmm calculator

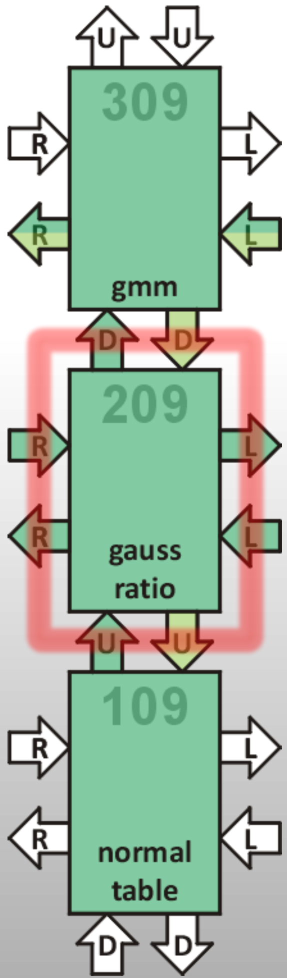
normal distribution lookup table




```
10109 +node 10109 /ram up /b 2F init /p
116 -256 + 3FFFF and 1 /stack

reclaim 10109 node 0 org data 24 - 0
1 , 2 , 5 , 9 , 10 ,
1C , 30 , 4E , 7C , BF ,
11E , 1A0 , 24D , 32B , 43E ,
586 , 6FF , 89E , A52 , C07 ,
DA1 , F05 , 101A , 10C9 , 1105 ,

abs -if - 1 . + then ;
p mn-mp over . + abs -25 . +
-if - a! @ ; then dup or ;
go 3F a! @ 9 for
255 for pop dup push p !b next @b next go ;
init 2F 3F a! ! go ;
```

gmm calculator

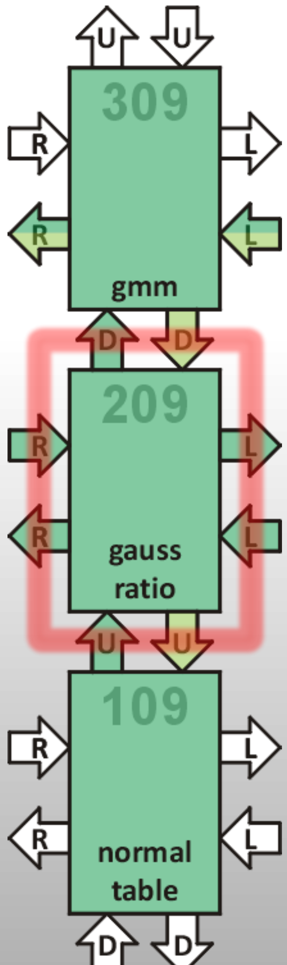
responsibility ratio calculation



$$r_j = \frac{\alpha_j N(x \mid \mu_j, \sigma^2_{fixed})}{\sum_{k=1}^{K} \alpha_k N(x \mid \mu_k, \sigma^2_{fixed})}$$

gmm calculator

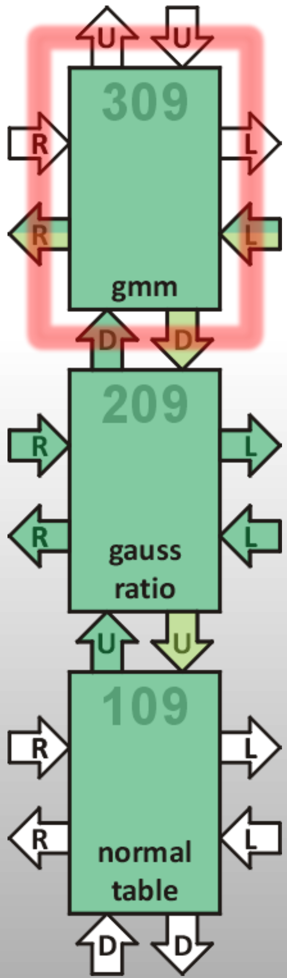
responsibility ratio calculation




```
10209 +node 10213 /ram 20 go /p

odd cols reclaim 10213 node 0 org
aph @p drop @p ; aph! !p ; .. 1555 1/12 ,
1+ 1 . + ;
*.r ab-aa*b a! dup dup or
15 for +* unext a -if drop 1+ ;
then drop ; +cy
/. hd-rq clc a! dup dup or 15 for begin
dup . + push dup . + dup a . + -if
drop pop swap next dup . + ; then
over or or pop next dup . + ; -cy
rj pp-r left a! @ . + a push right a! !
@ dup pop a! ! /. ;
go 20 9 for 255 for up b! aph @b *.r dup rj
down b! !b next
@b @b aph! up b! !b next
1555 aph! go ;
```

gmm calculator

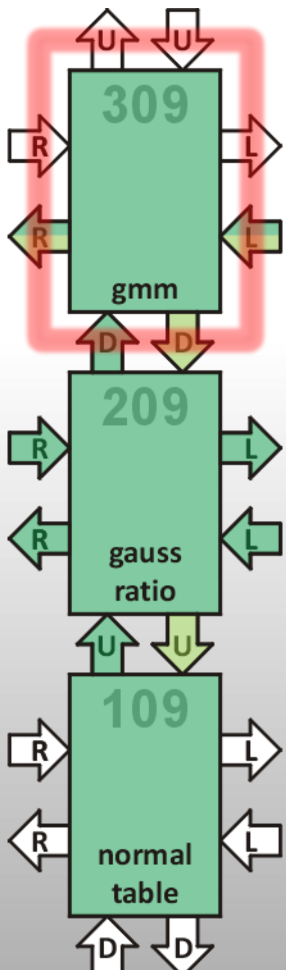
EM calculation



$$\hat{\alpha}_j = \sum_{i=1}^{N} r_j\, y_i \qquad \hat{\mu}_j = \frac{\sum_{i=1}^{N} r_j\, y_i\, x_i}{\sum_{i=1}^{N} r_j\, y_i}$$

gmm calculator

EM calculation




```
10309 +node 10313 /ram 18 go /p

odd cols reclaim 10313 node 0 org
ry @p drop @p ; ry! !p ; 0 ,
ry+ ry . + ry! ;
ryx @p drop @p ; ryx! !p ; 0 ,
ryx+ ryx . + ryx! ;
1+ 1 . + ;
*.r ab-aa*b a! dup dup or
15 for +* unext a -if drop 1+ ; then drop ;
* ab-aa*b a! dup dup or 8 for +* unext ;
go 18 dup or dup ry! ryx!
255 for down a! @ r
left b! @b dup right b! !b *.r ry
pop dup push 2* 2* 2* 2* 2* 2* 2* 2* 2*
over * 2/ 2/ 2/ ryx
ryx+ drop ry+ next
dup or ryx ry 2/ 2/ 2/ - 1+ --u/mod mu
- dup 1+ down a! ! ry alpha dup ! 1F5 r-l- a!
- ahead begin swap ! then . .. . @ -until
- ! ! ! go ; 40
```

380 μs

1.24 μs

61 μs

DEMO #2

echo processing

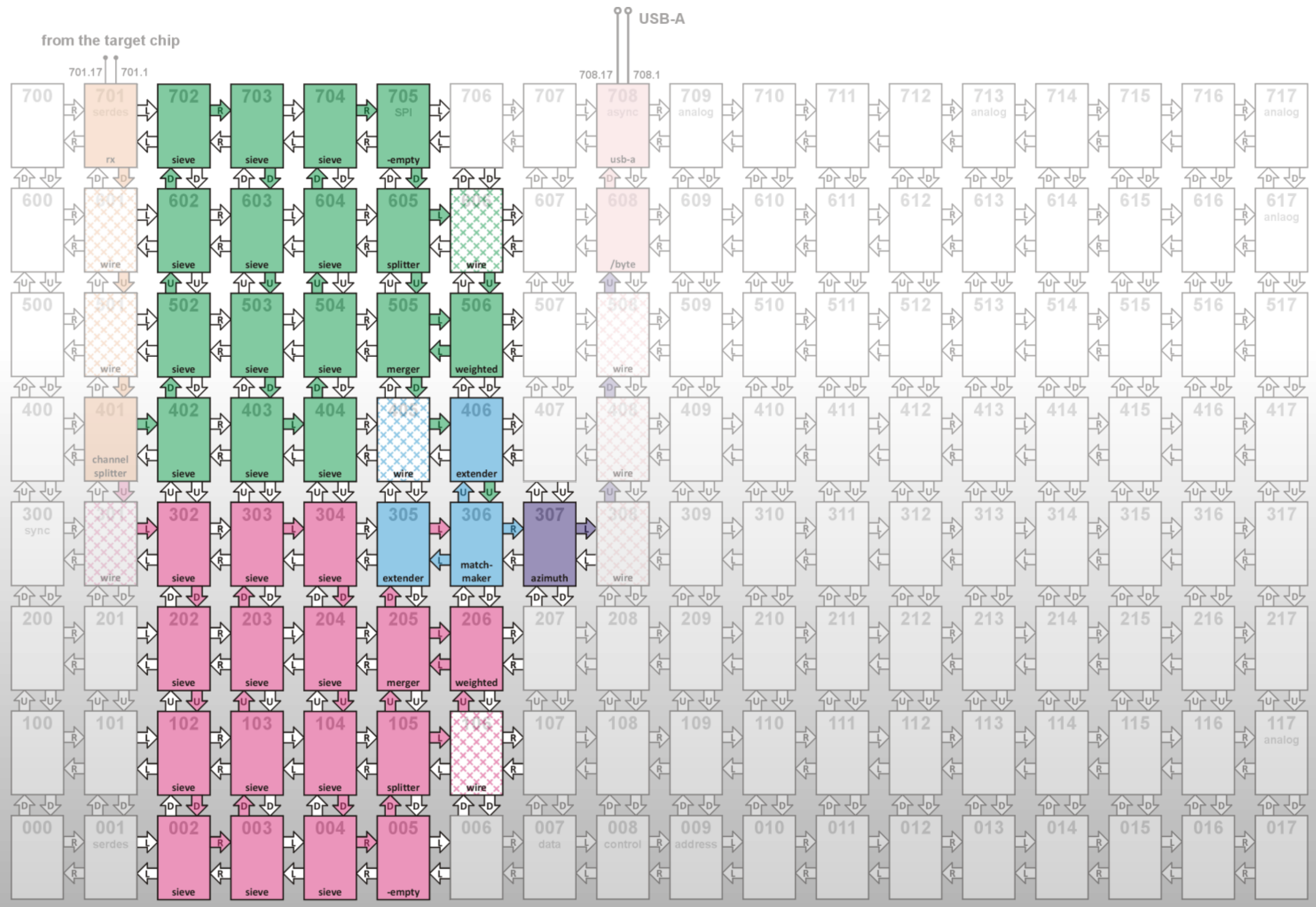
[link to demo #2 video on YouTube]

IMPLEMENTATION

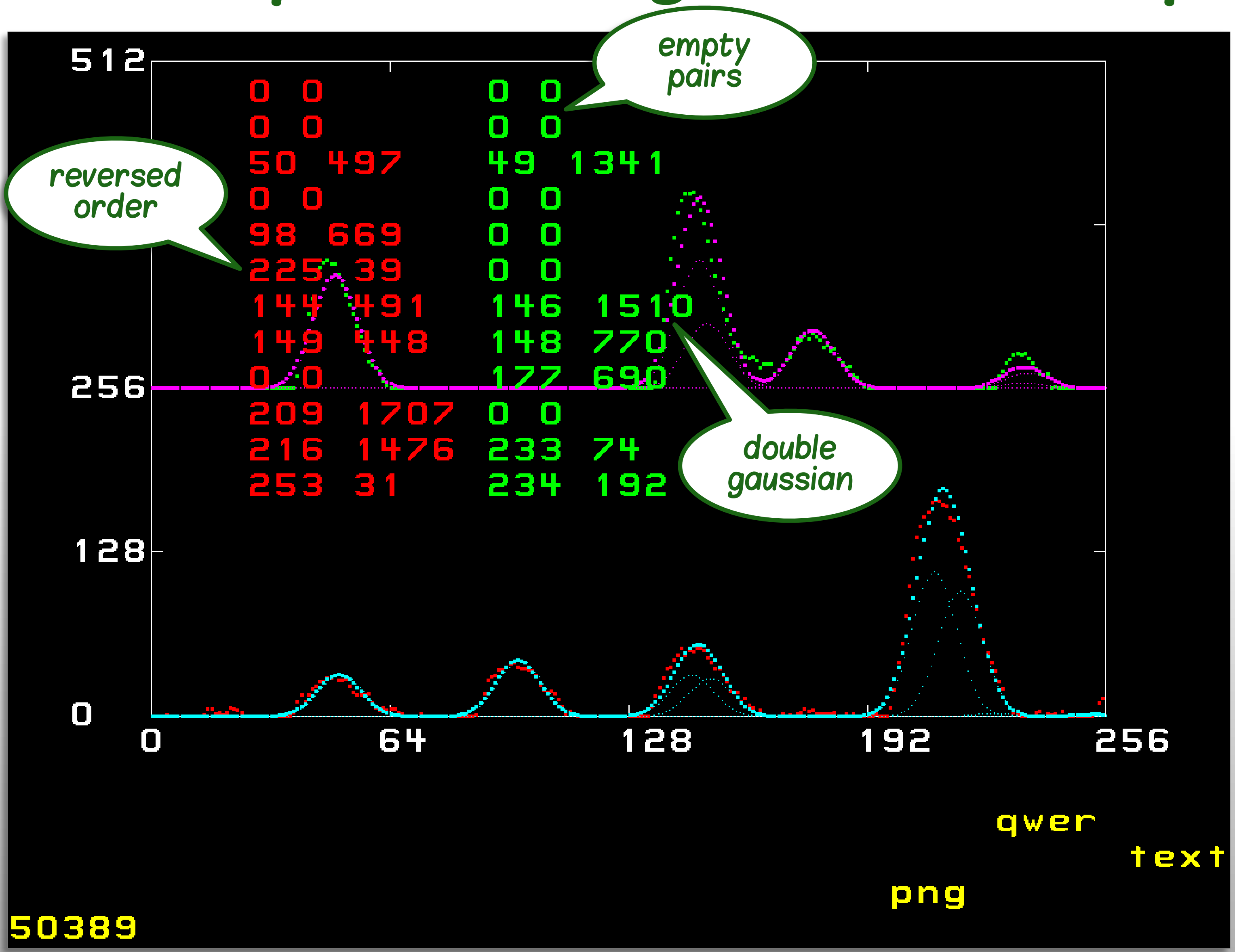
part III
azimuth & distance determination

host chip

azimuth & distance determination



echo processing module output

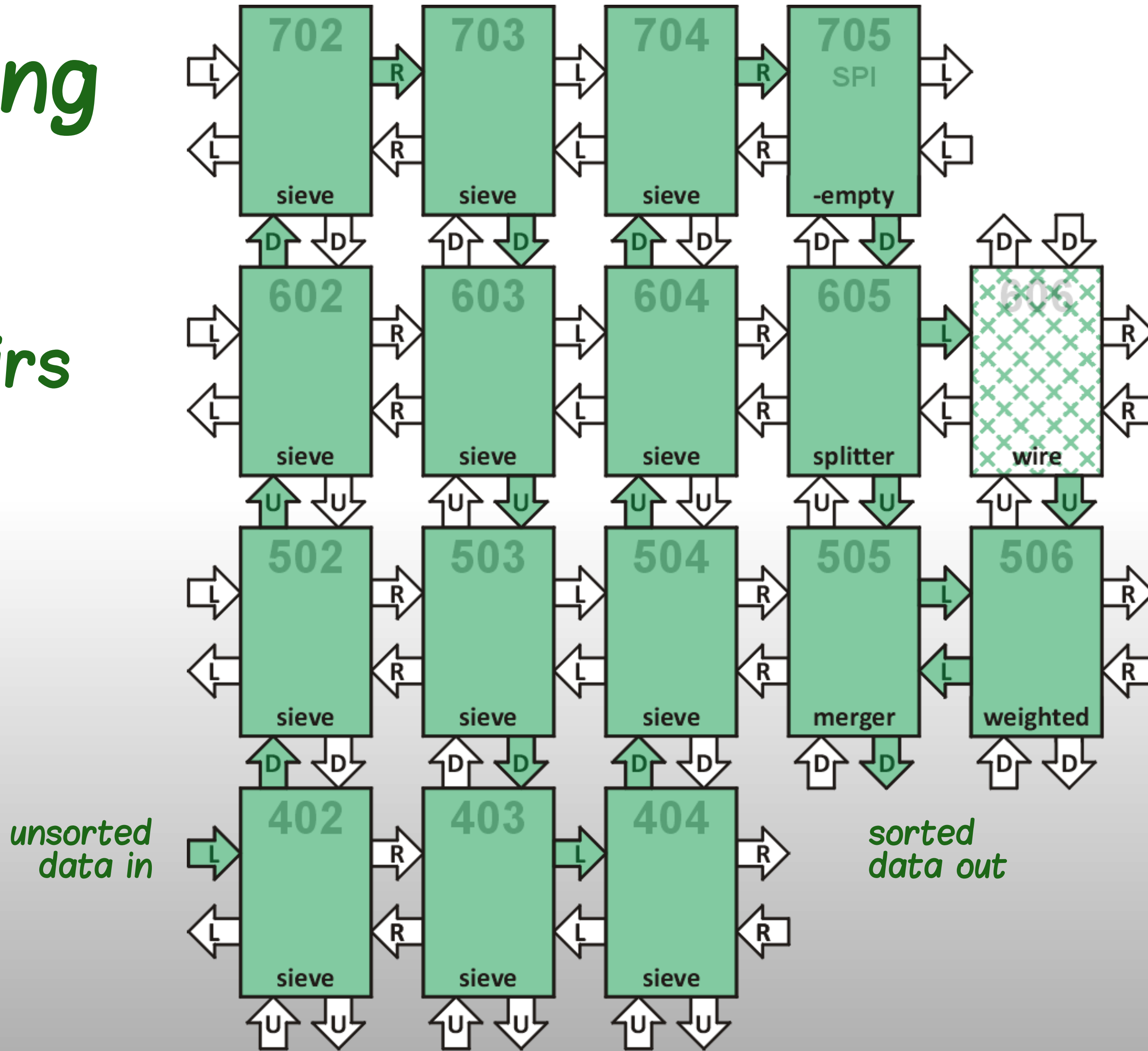


post-processing module

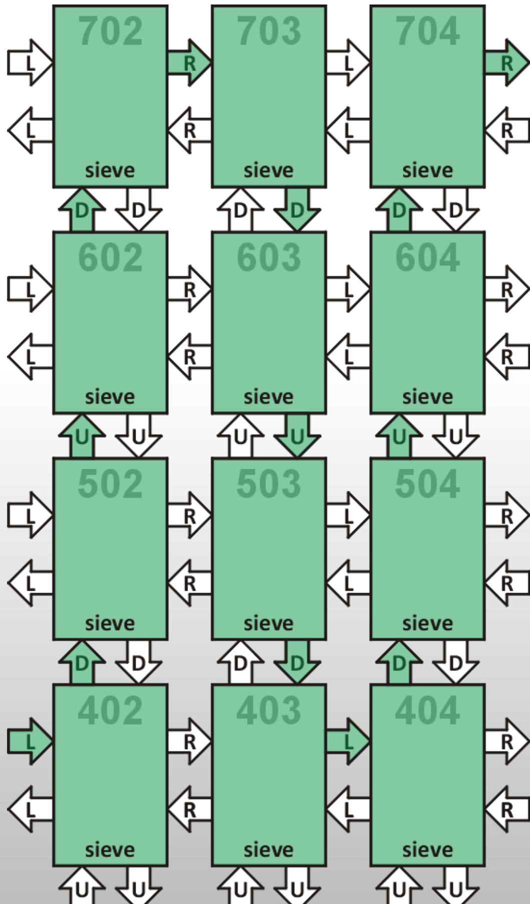
1) sorts pairs according to increasing mu
2) removes all zero pairs
3) merges double gaussians

post-processing module

- sieves
- empty pairs removal
- merger



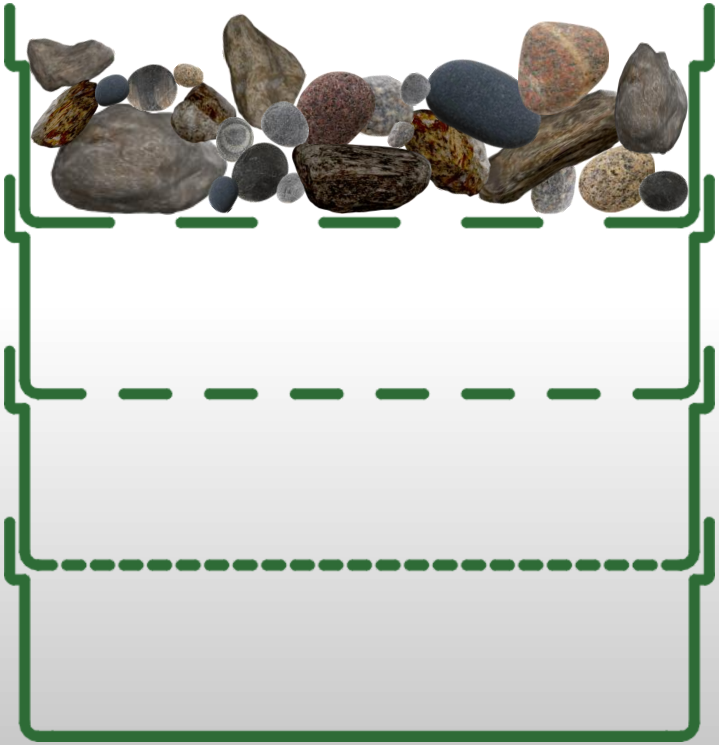
sieves




```
402 +node 402 /ram left /a down /b 11 go /p

reclaim 402 node 0 org
le mn-f - . + ;
-fall am-am dup push @ @ pop over le
-if drop push push over !b !b pop pop ;
then drop over !b !b drop ;
flush am !b !b ;
fill -am @ @ 10 for -fall next ;
go 11 fill flush go ;
```

sieve analysis


wikimedia commons

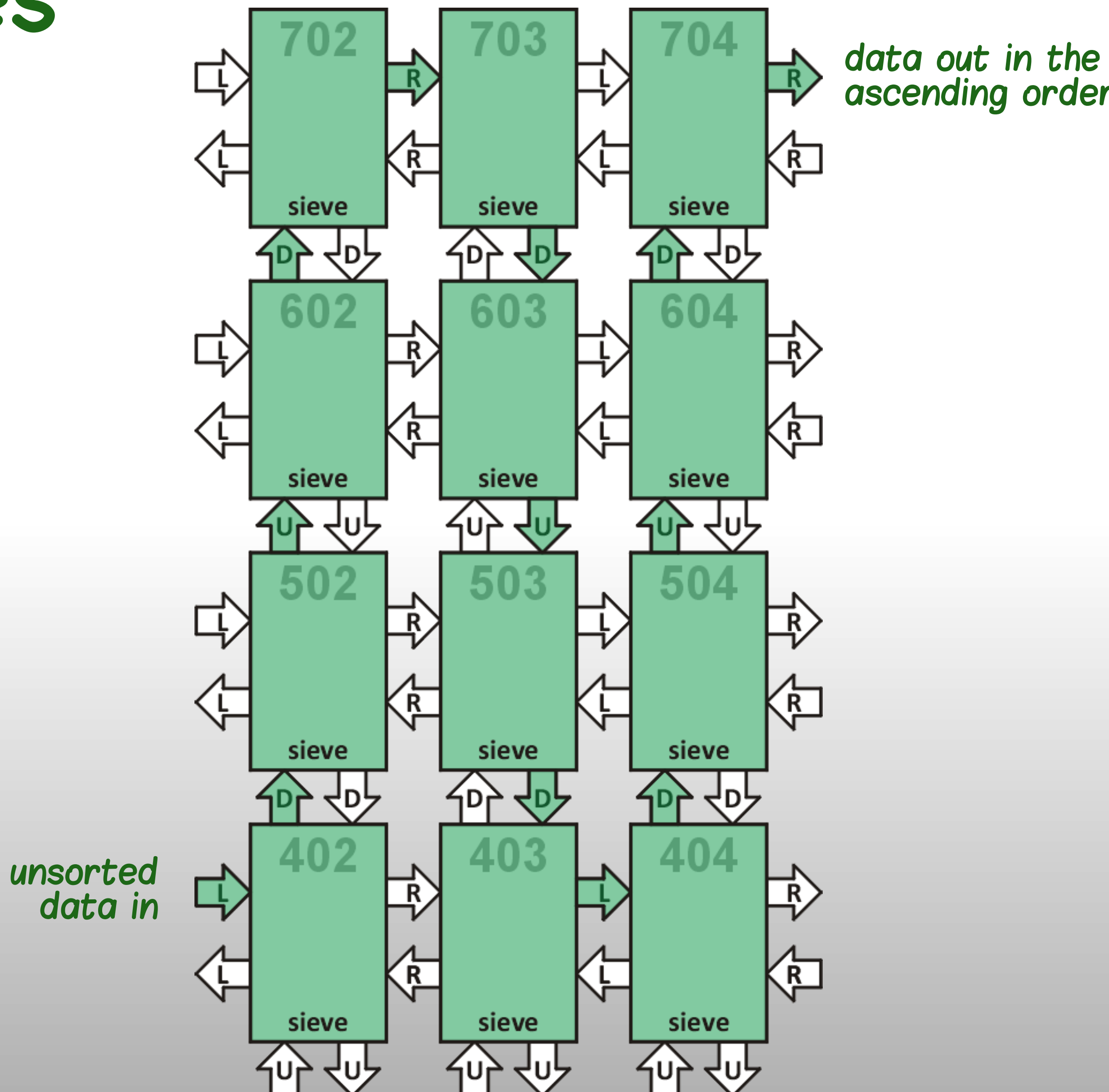


sieve analysis

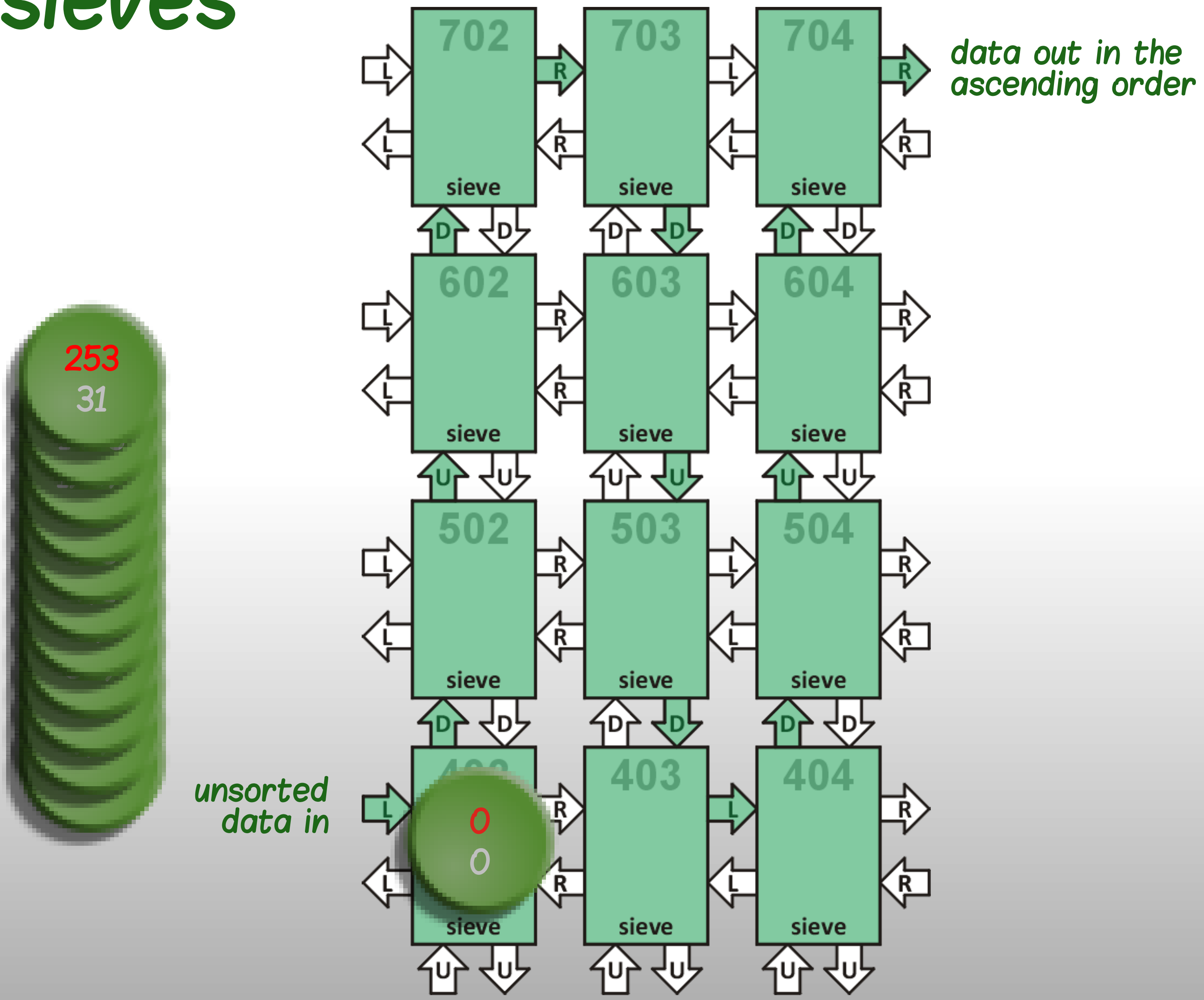
wikimedia commons

sieves


253
31

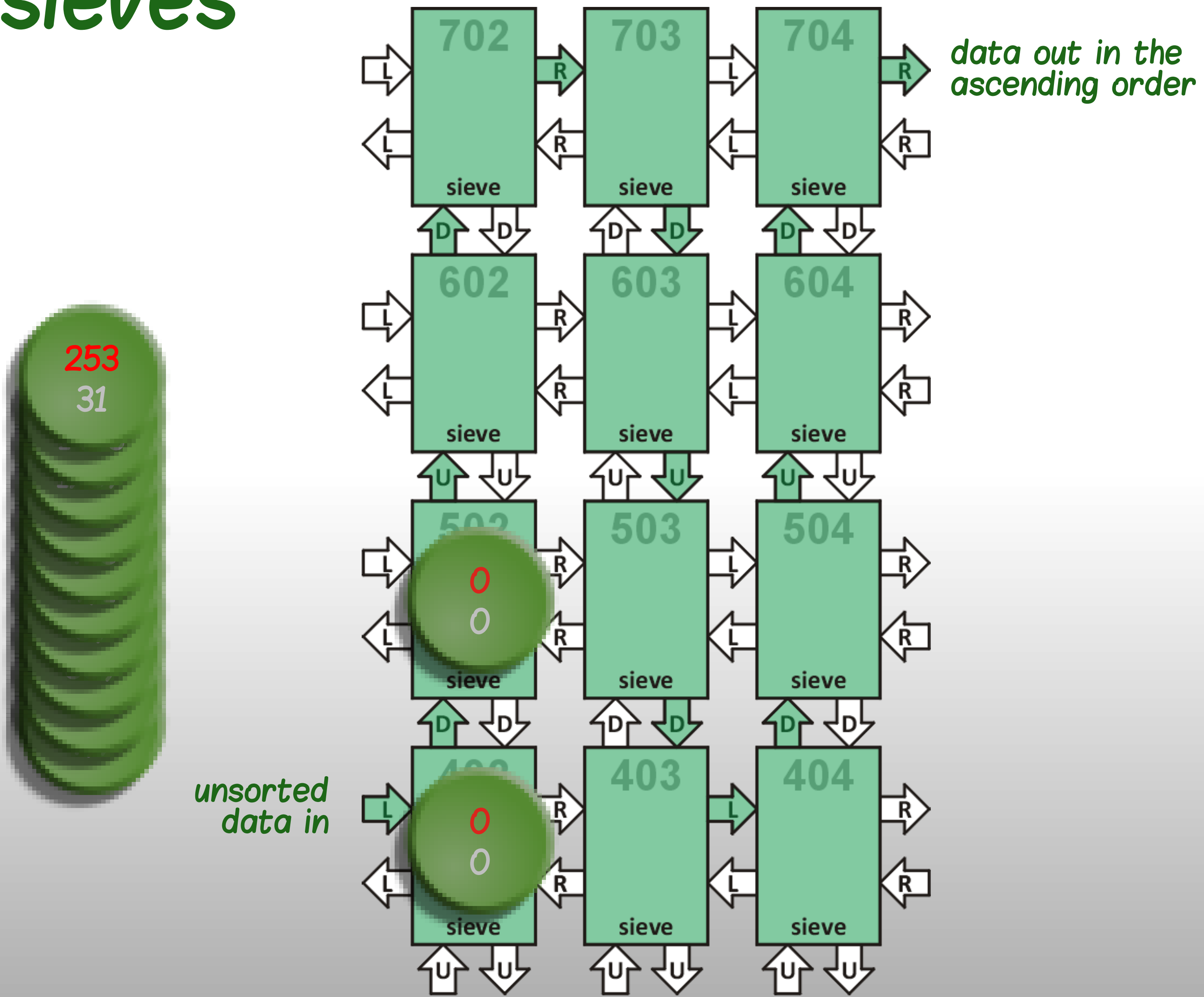

data out in the ascending order
702
703
704
602
603
604
502
503
504
402
403
404
sieve
L
R
D
U
unsorted data in

sieves

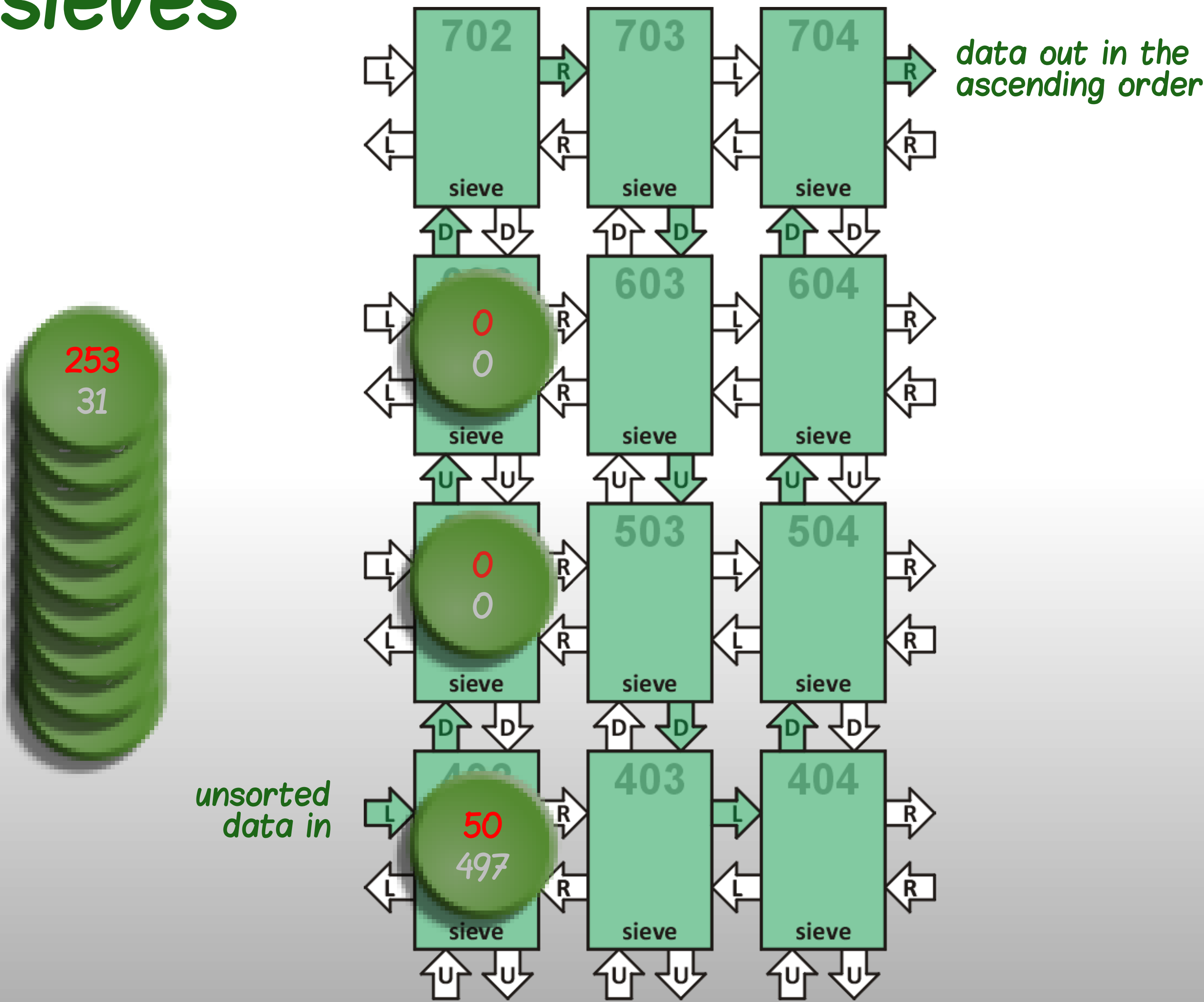


253
31
unsorted data in
0
0
702
703
704
602
603
604
502
503
504
403
404
sieve
data out in the ascending order

sieves

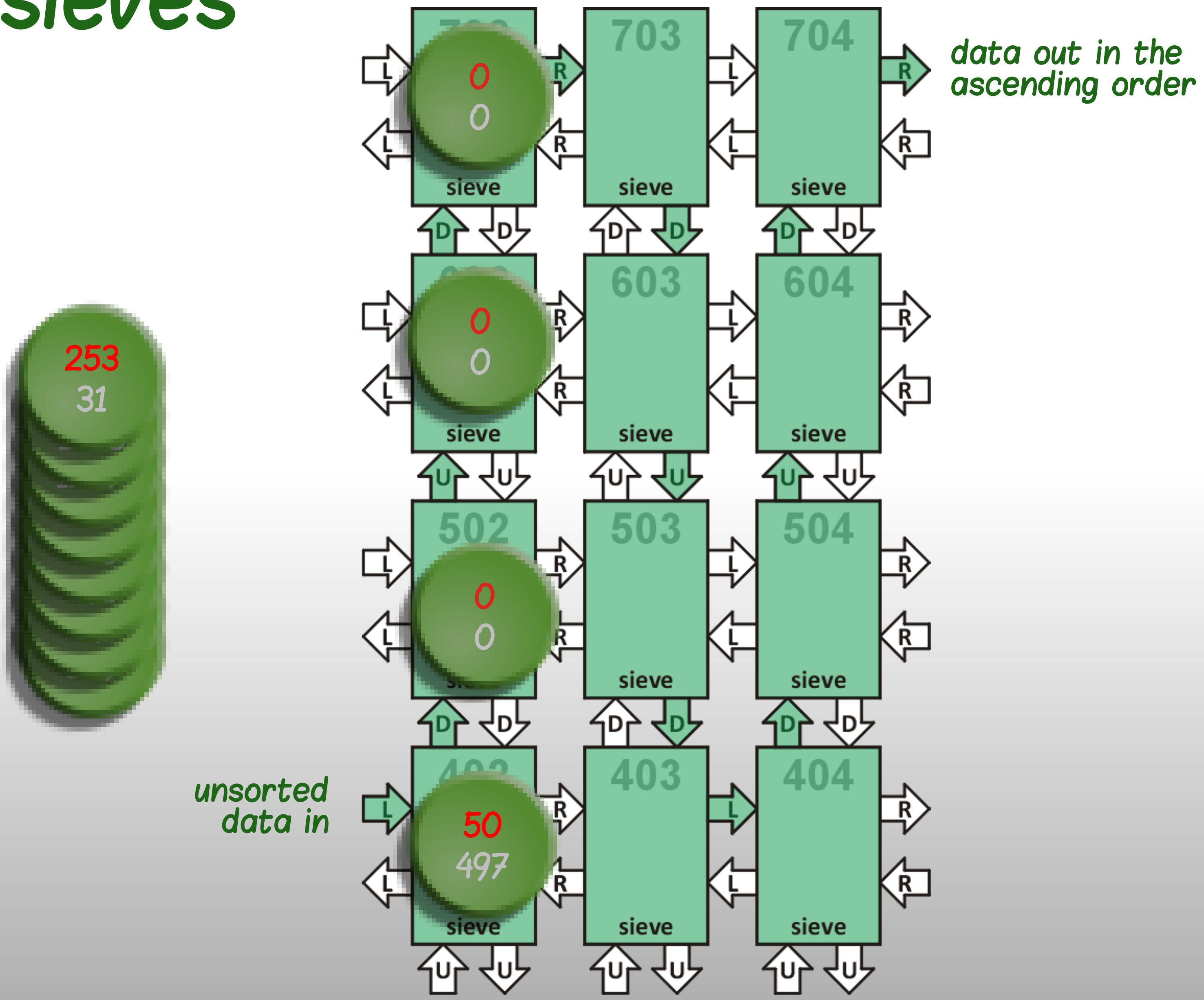

253
31
unsorted
data in
702
703
704
602
603
604
503
504
403
404
sieve
L
R
D
U
0
0
0
0
data out in the
ascending order

sieves



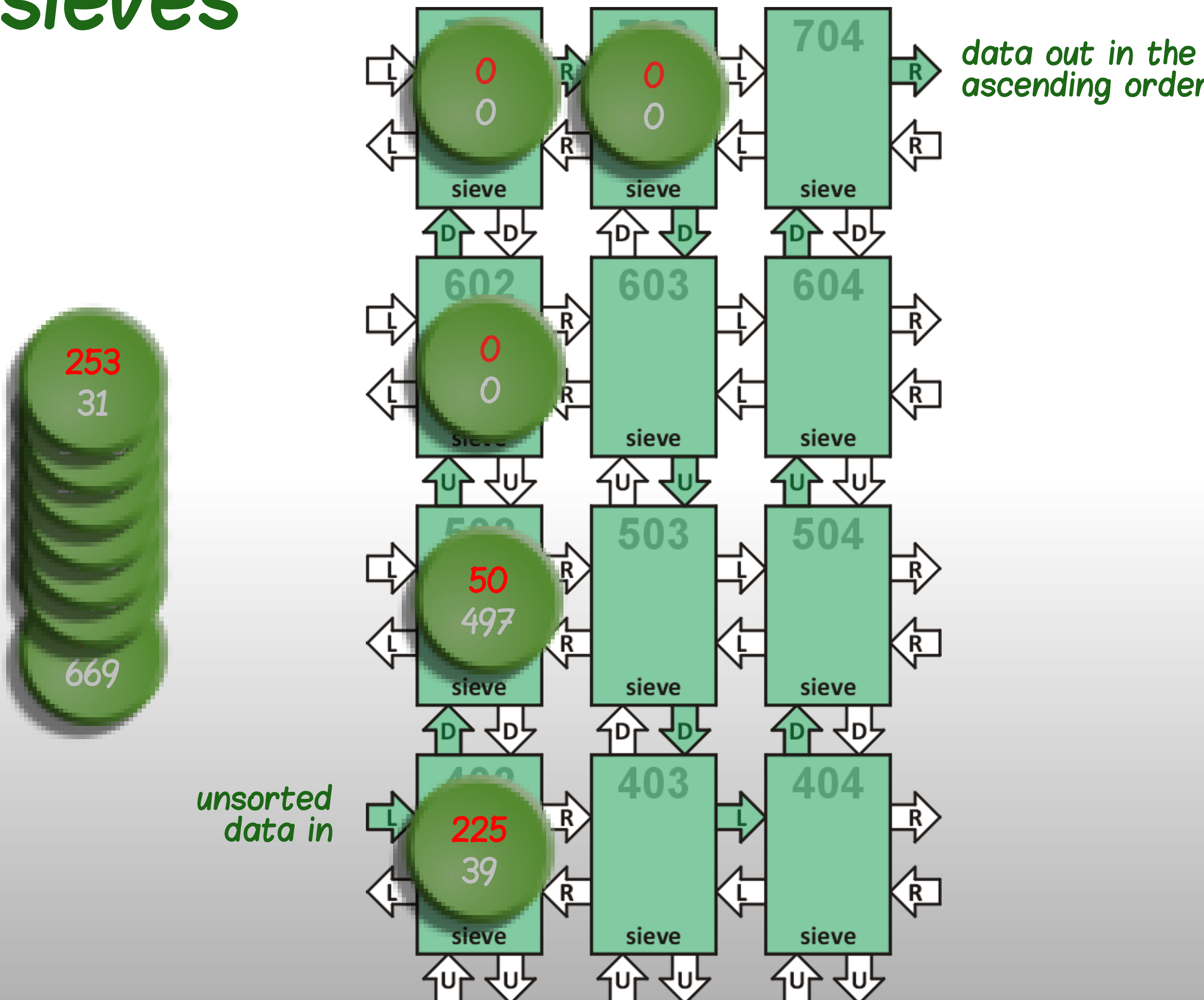
253
31
unsorted
data in
702
703
704
603
604
503
504
403
404
sieve
0
0
0
0
50
497
data out in the
ascending order

sieves

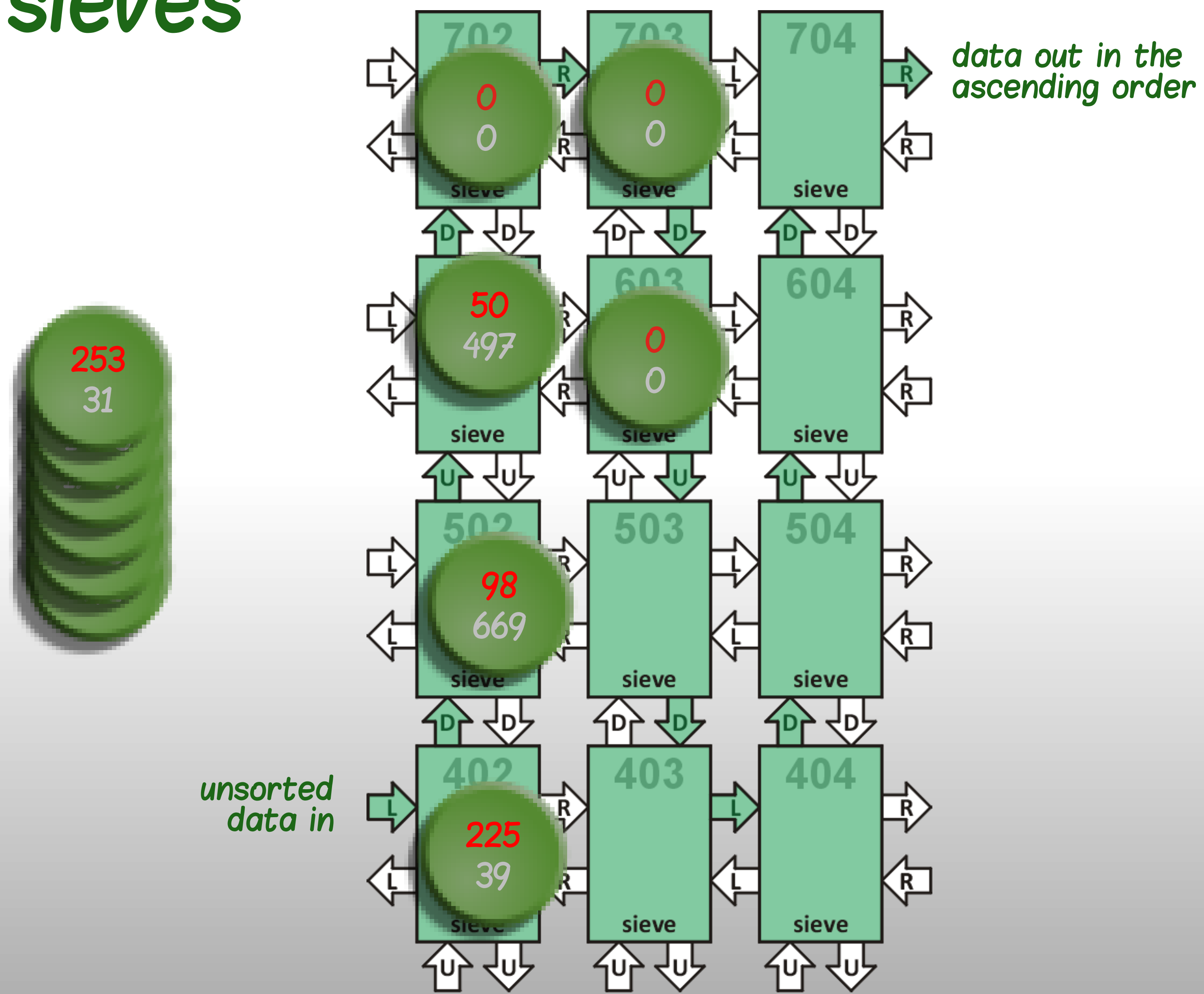


253
31
unsorted
data in
data out in the
ascending order
703
704
603
604
502
503
504
403
404
sieve
0
0
50
497
L
R
D
U

sieves

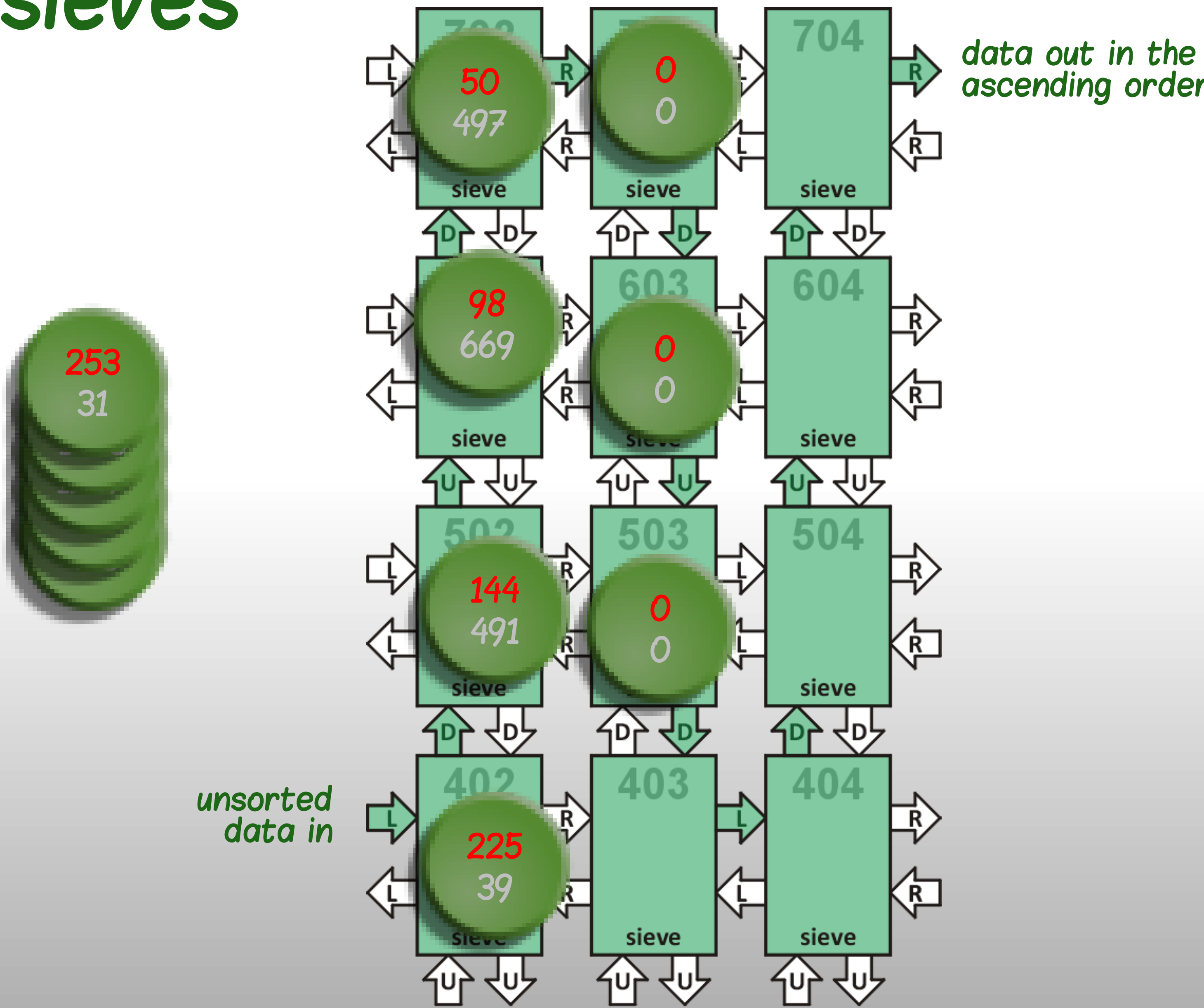

data out in the ascending order
unsorted data in
253
31
669
0
0
0
0
0
0
50
497
225
39
704
602
603
604
503
504
403
404
sieve
L
R
D
U

sieves



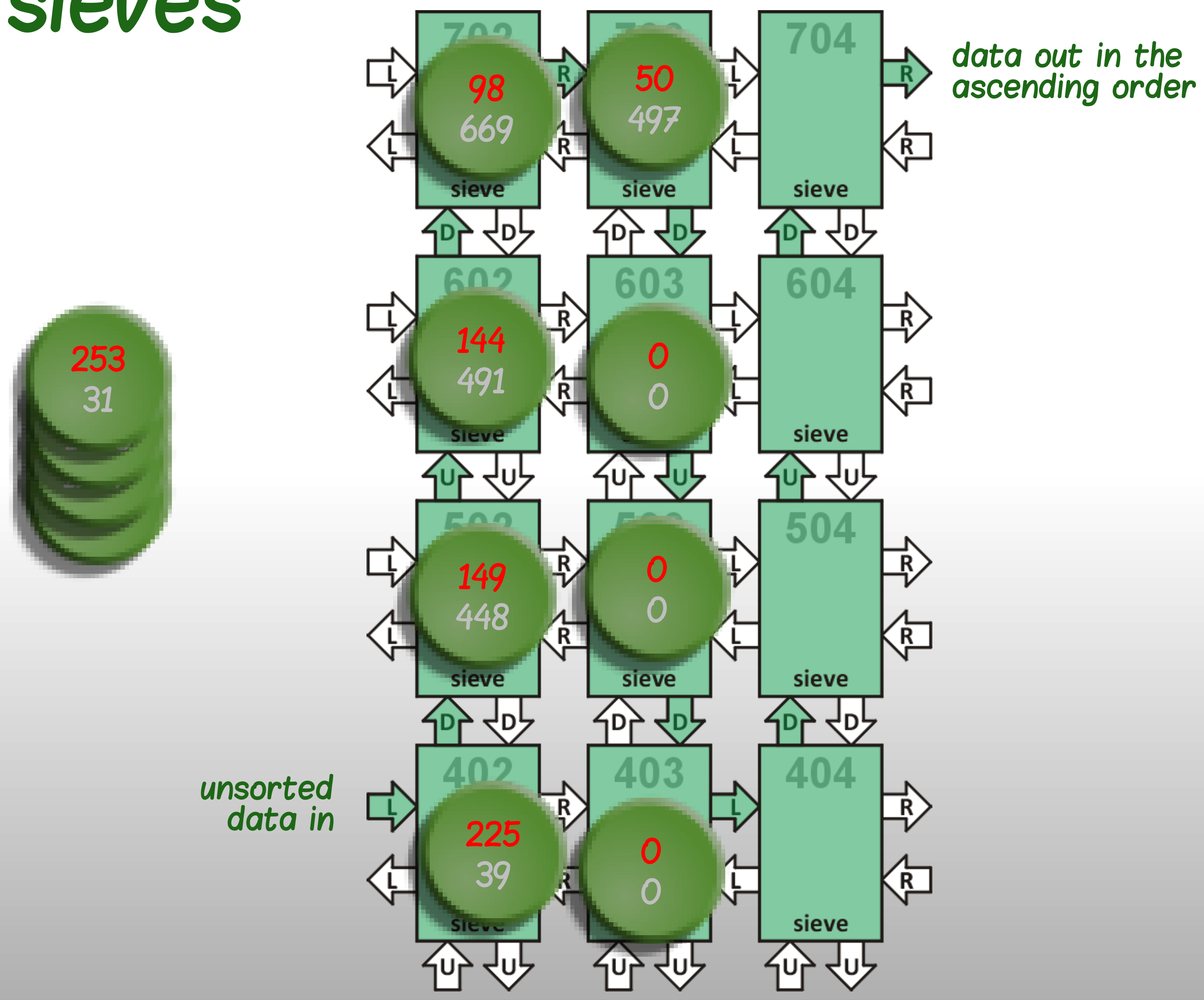
data out in the ascending order
unsorted data in
702
703
704
603
604
502
503
504
402
403
404
sieve
L
R
D
U
0 0
0 0
50 497
0 0
98 669
225 39
253 31

sieves



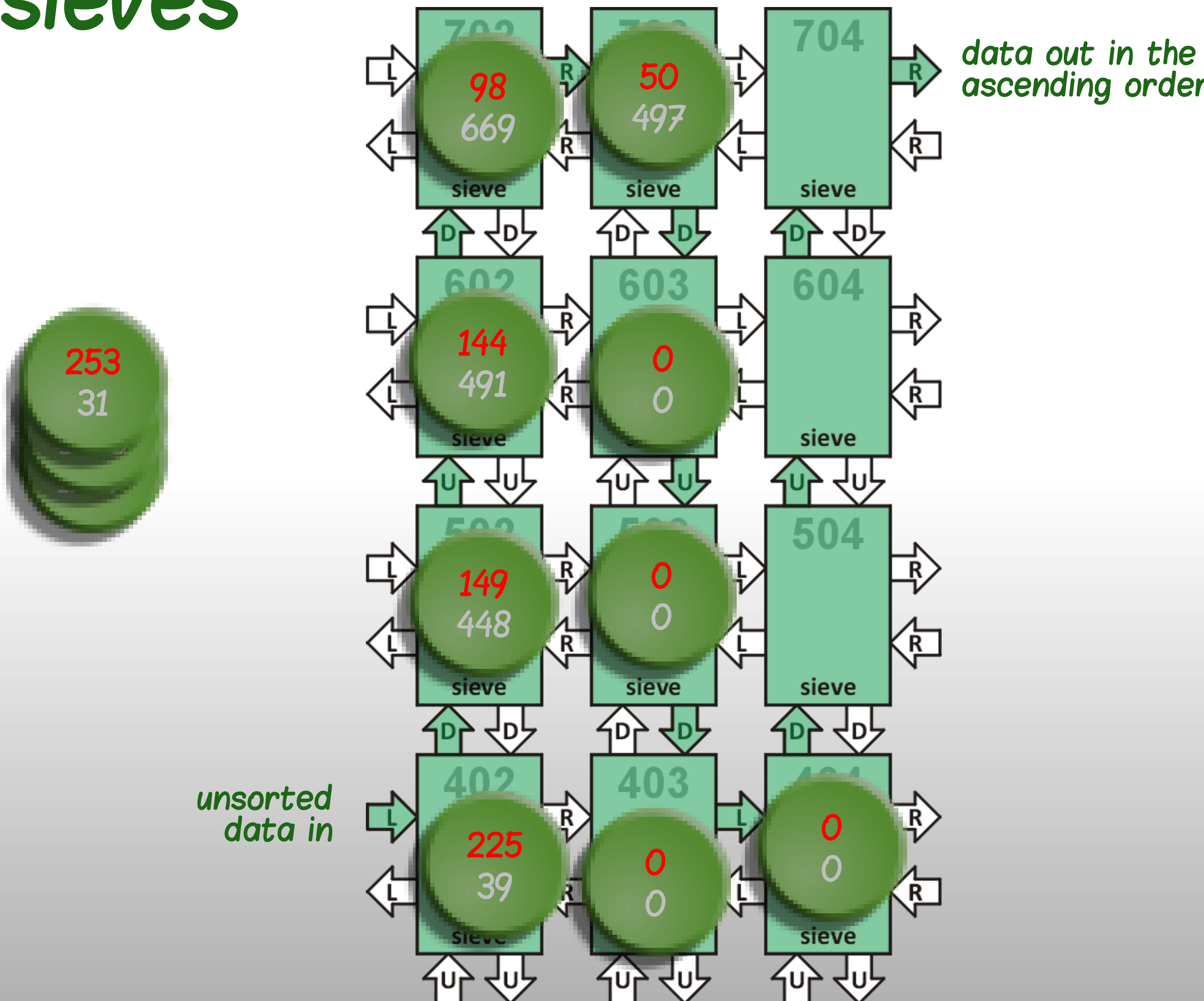
data out in the ascending order
unsorted data in
704
603
604
503
504
403
404
sieve
L
R
D
U
253
31
50
497
0
0
98
669
0
0
144
491
0
0
225
39

sieves

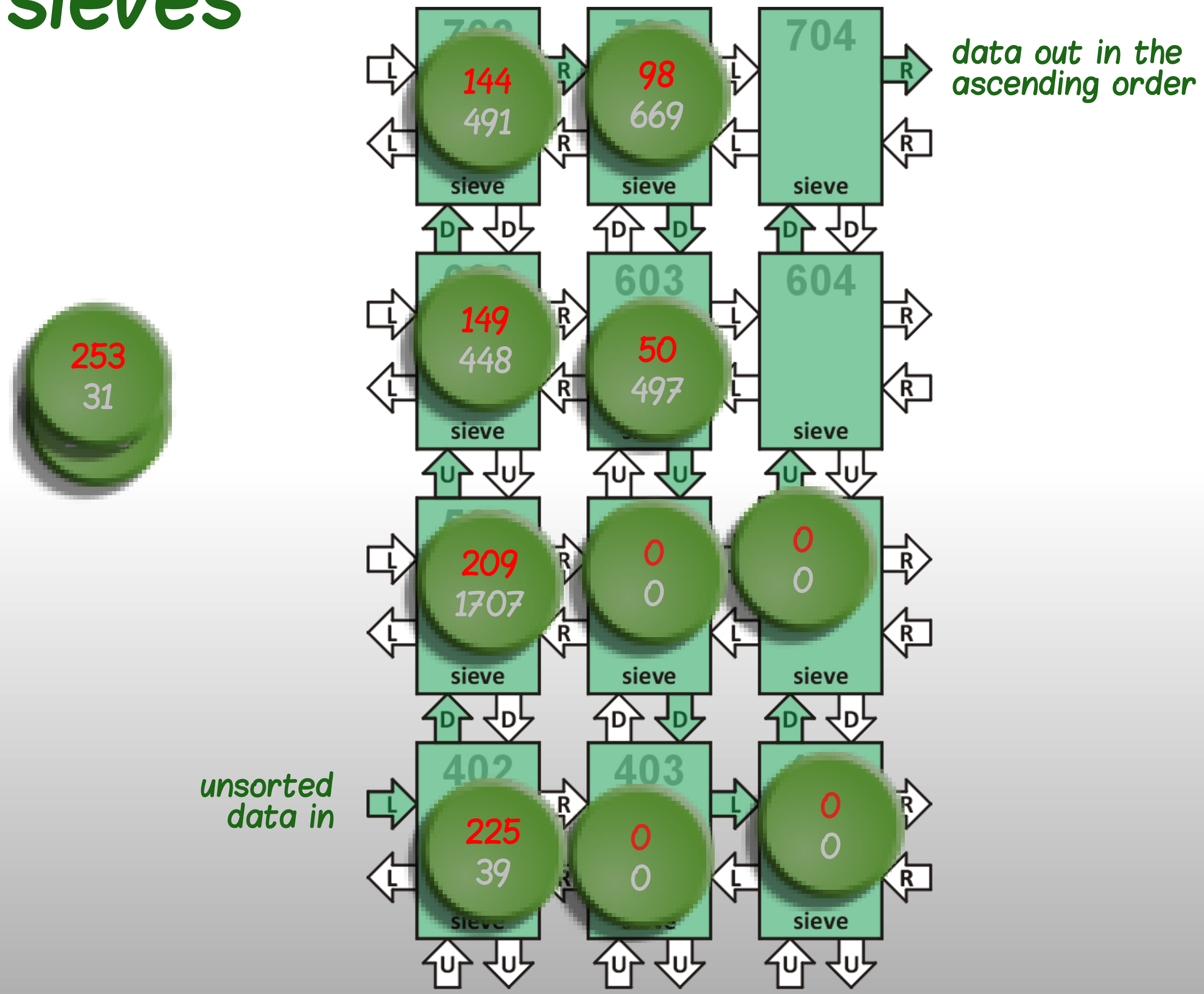


253
31
702
98
669
sieve
703
50
497
sieve
704
sieve
602
144
491
sieve
603
0
0
604
sieve
502
149
448
sieve
503
0
0
sieve
504
sieve
402
225
39
sieve
403
0
0
404
sieve
L
R
D
U
data out in the ascending order
unsorted data in

sieves

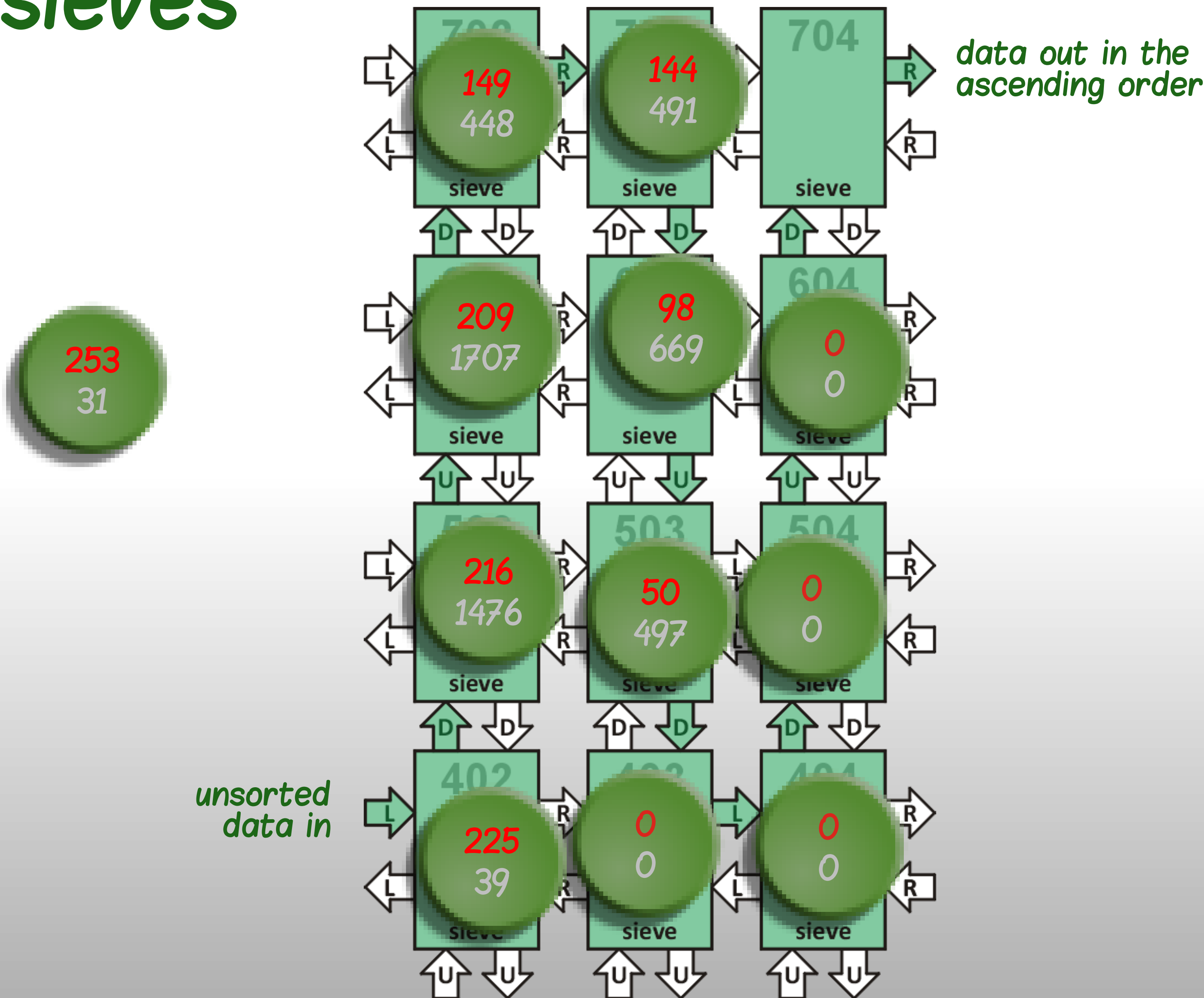

data out in the ascending order
unsorted data in
253
31
98
669
50
497
704
603
604
504
403
144
491
149
448
225
39
0
0
sieve
L
R
D
U

sieves

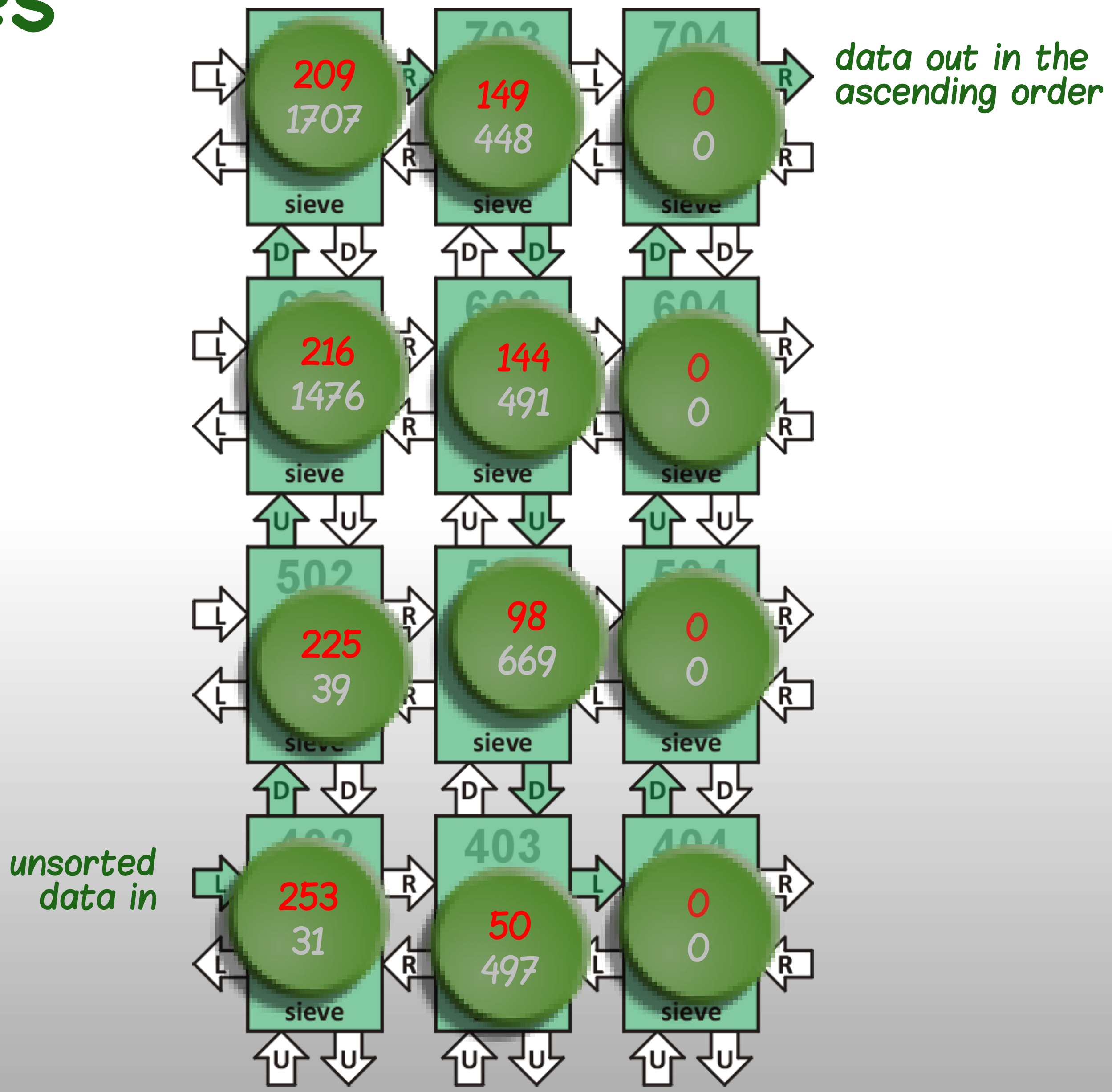


data out in the ascending order
unsorted data in
704
603
604
402
403
sieve
144
491
98
669
149
448
50
497
209
1707
225
39
253
31
0

sieves

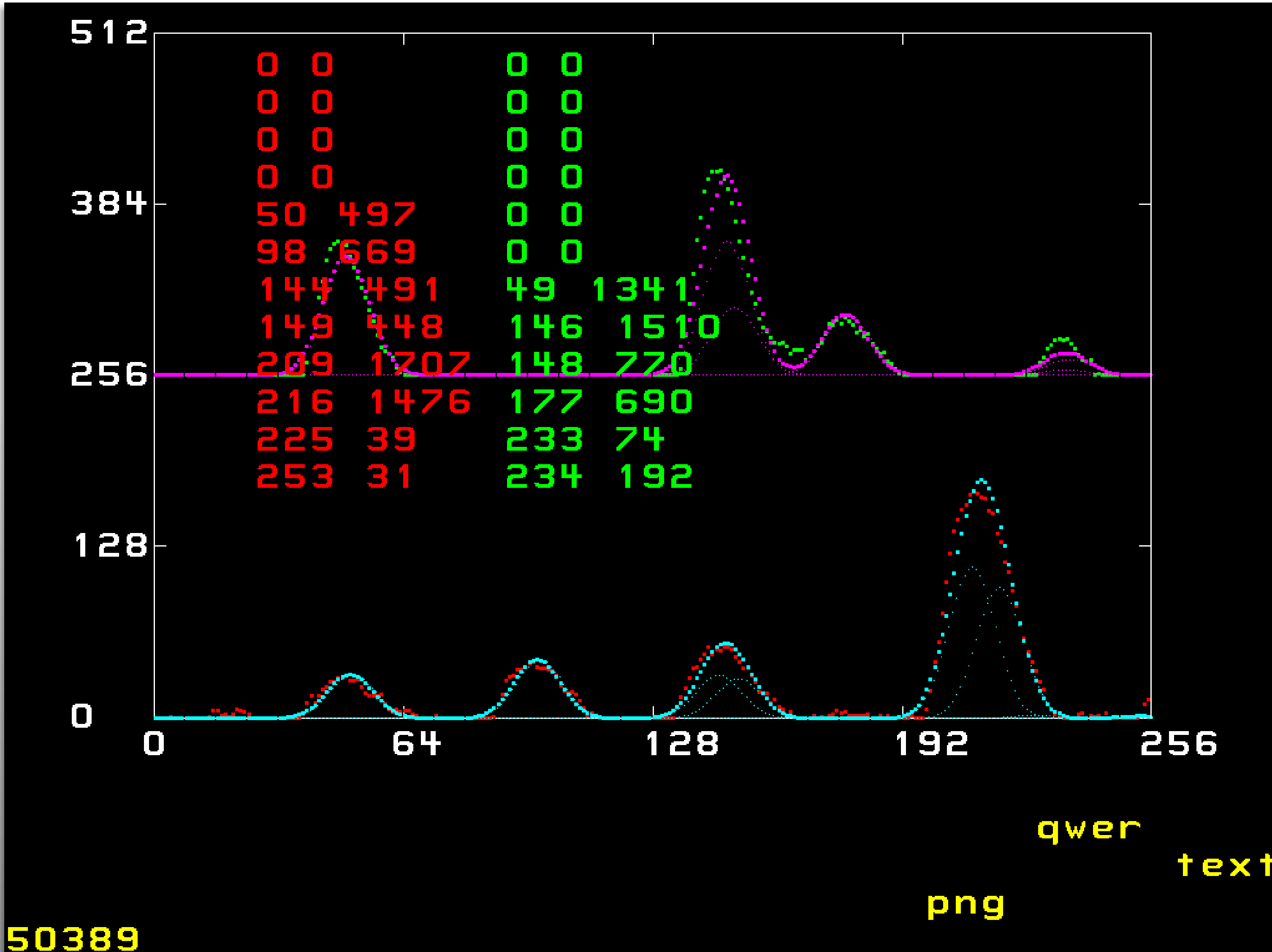

data out in the ascending order
unsorted data in
704
604
503
504
402
sieve
149
448
144
491
209
1707
98
669
0
0
253
31
216
1476
50
497
225
39
L
R
D
U

sieves

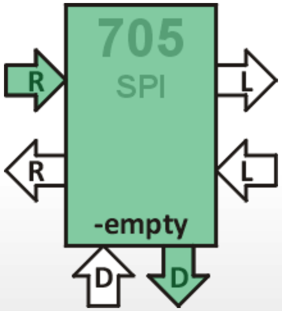


data out in the ascending order
unsorted data in
209
1707
149
448
0
0
216
1476
144
491
0
0
502
225
39
98
669
0
0
403
253
31
50
497
0
0
sieve
L
R
D
U

sieves output



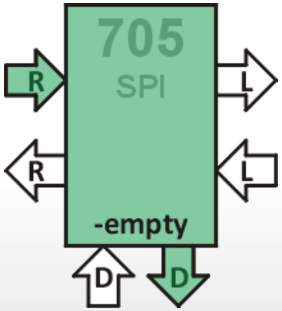
empty pairs removal



removes zero pairs

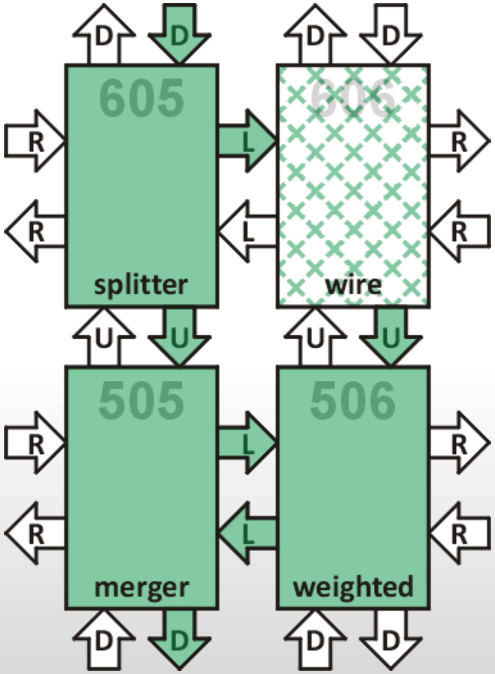
ends data with -1 flag

empty pairs removal




```
705 +node 705 /ram right /a down /b 0 go /p

reclaim 705 node 0 org
go 11 for @ @ over over . +
if drop !b !b then next
dup or - !b go ;
```

merger



$a_n \; m_n \; a_{n+1} \; m_{n+1} \ldots$

$m_{n+1} - m_n \leqslant 6$

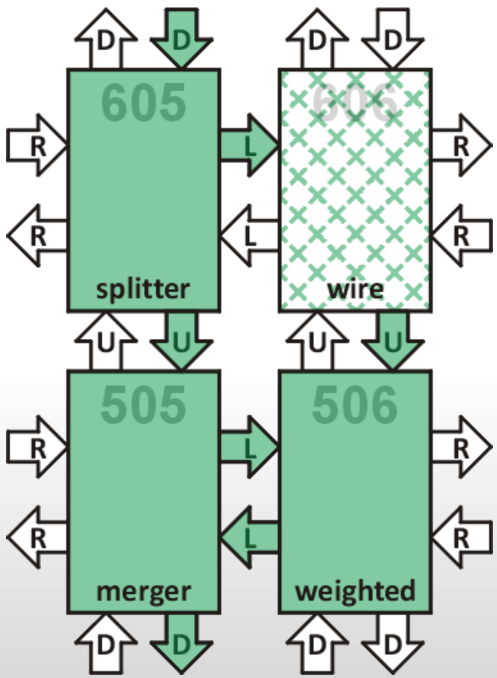
$m_{n+1} - m_n > 6$

$$m = \frac{m_n a_n + m_{n+1} a_{n+1}}{a_n + a_{n+1}}$$

m_n, a_n out

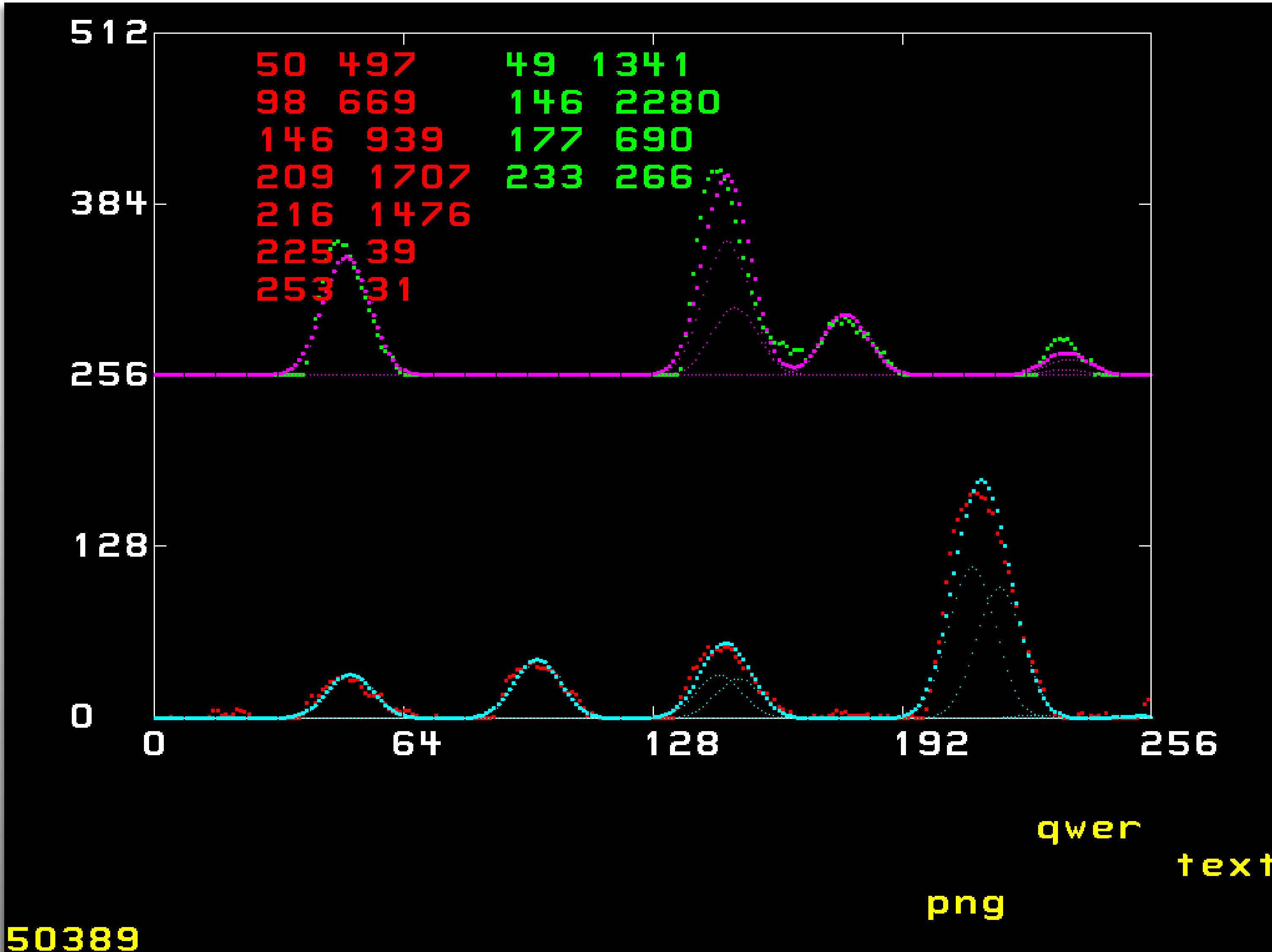
$a = a_n + a_{n+1}$

merger

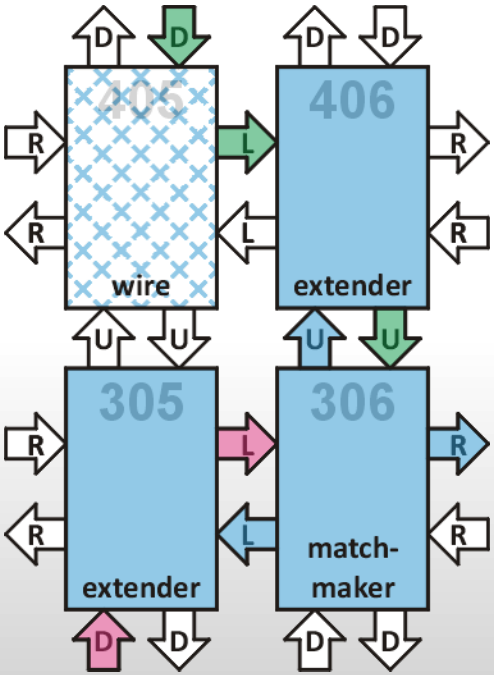



```
605 +node 605 /ram 135 -dl- /a up /b 0 go /p
506 +node 506 /ram left /b 0 one /p
505 +node 505 /ram 105 -d-u /a left /b A
first /p

reclaim 605 node 0 org
go @ -if !b go ; then ! @ !b go ;

reclaim 506 node 0 org
one up a! @ --l- ; 1+ 1 . + ;
* ab-a*b a! 17 push dup dup or
begin +* unext a ; +cy
+d clc a! push a . + a! pop . + a ; -cy
weigh aa-a over over . + dup push push
@b * push push drop @b * pop pop +d
pop - 1+ --u/mod !b pop --l- ;

reclaim 505 node 0 org
near? ab-abf over push dup pop
ba - . + -6 . + ;
blend @p !b .. weigh ; !b !b @b ;
solo over @p !b .. !p one ; ! @b ! ;
first 0A @ -if ! first ; then
follow @ -if solo ! first ; then
@p !b .. one ;
near? -if drop blend follow ; then
drop @p !b .. over !p .. over ! @b !
follow ;
```

merger output



matchmaker module



$abs(m_L - m_R) \leqslant 6$

$m = (m_L + m_R) / 2$

m, a_L, a_R out

$abs(m_L - m_R) > 6$

smaller(m_L, m_R) out

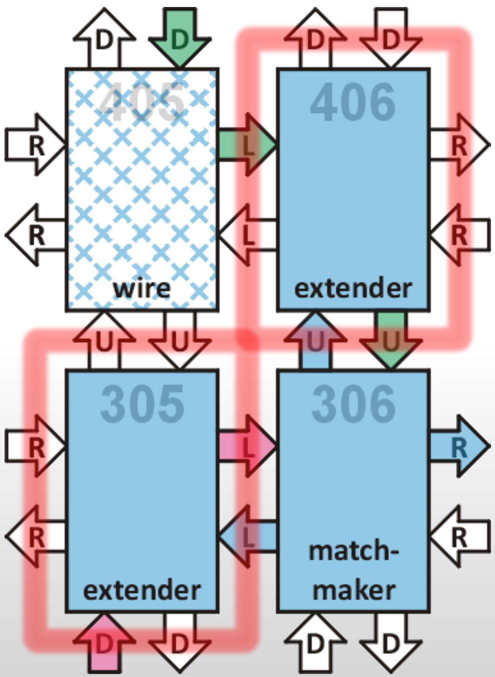
corresponding a out

other channel $a = 0$ out

$m_L, a_L, 0$ or $m_R, 0, a_R$

matchmaker module

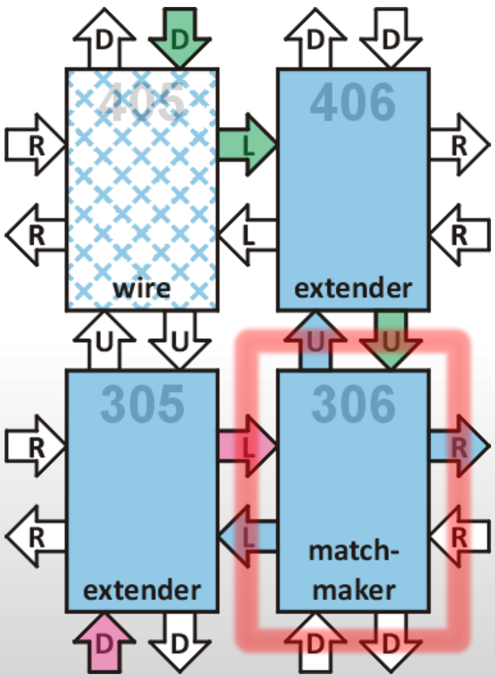
extender




```
406 +node 406 /ram up /a A init /p

reclaim 406 node 0 org
xqt @p @p push ; xqt! !p ; .. 0 ,
fin 10000 ! ;
nxt @b -if ' fin lit xqt! fin ; then ! ;
init 0A io b! ' nxt lit xqt!
go begin @b 800 lw and until left b! ---u ;
```

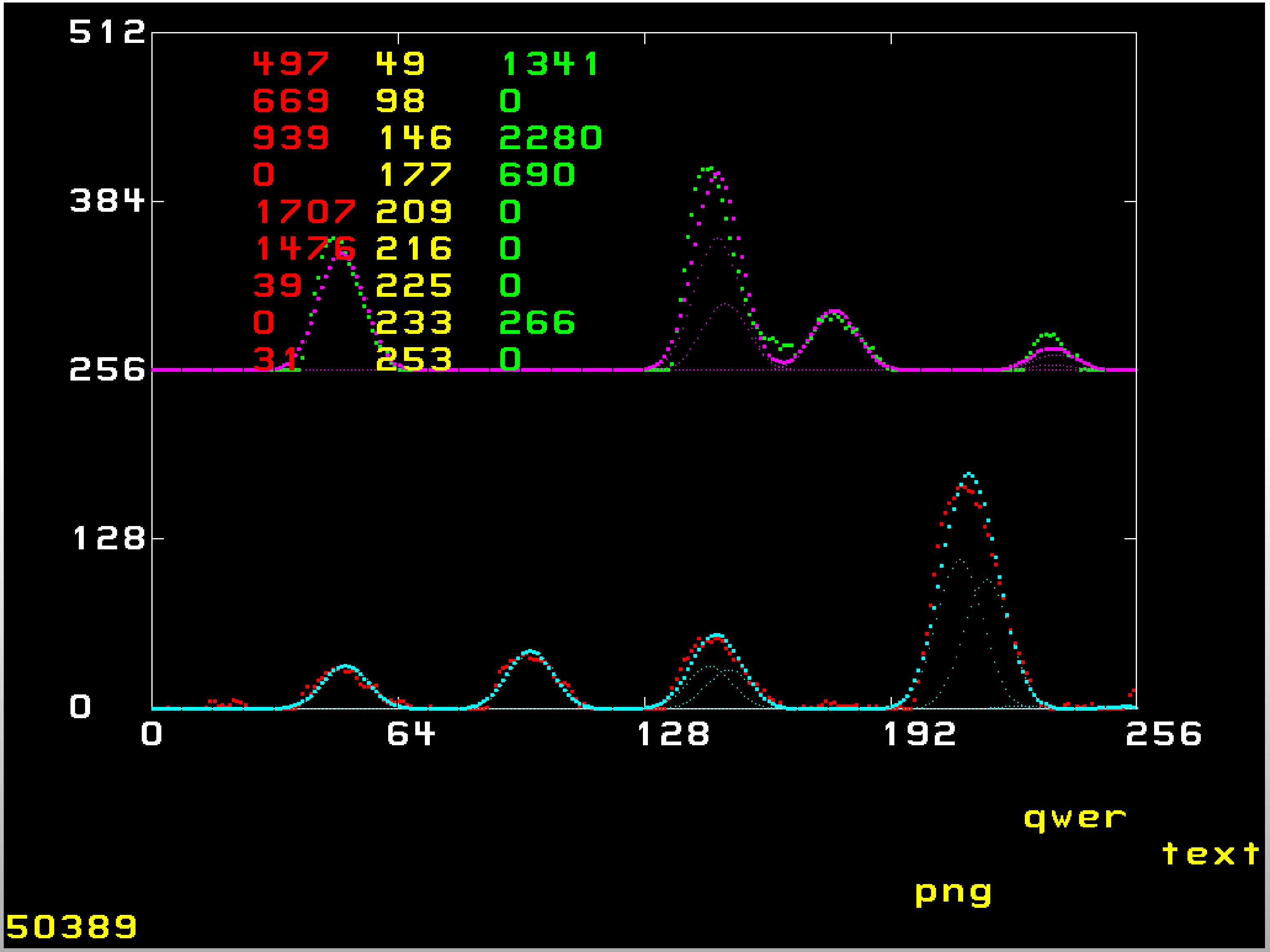
matchmaker module

matchmaker

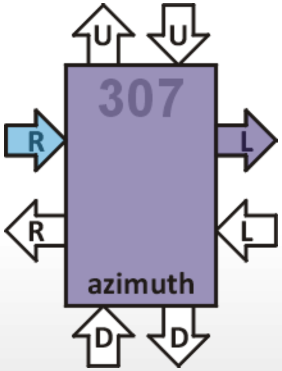



```
306 +node 306 /ram 1E start /p

reclaim 306 node 0 org
left@ left a!
get @p ! @ ; xqt ..
right@ up a! get ;
abs -if - 1 . + then ;
lt? ab-abf over over neg + ;
near? ab-abf lt? abs -6 . + ;
-emp? ab-abf over 2* over 2* . + ;
smlr mmf-m -if drop over !b right@ !b 0 !b
right@ over ;
then drop !b 0 !b left@ !b left@ ;
start 1E io b!
rdy? begin @b - dup 2/ 2/ and 400 and until
go right b!
begin right@ left@ upd -emp?
while drop near?
-if drop . + 2/ !b right@ !b left@ !b
else drop lt? smlr upd ; then
end then dup or - !b .. @p @p over @p init ..
145 up , 175 left , a! ! a! ! start ;
```

matchmaker module output



azimuth calculator



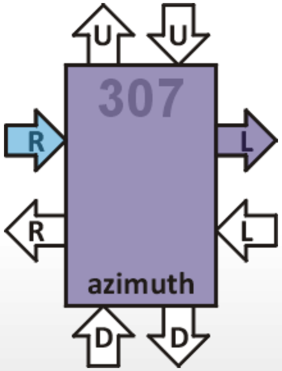
Michelson contrast

$$\text{azimuth} = \frac{a_R - a_L}{a_R + a_L}$$

$$-1 \leqslant \text{azimuth} \leqslant 1$$

$$\text{distance} \sim \mu$$

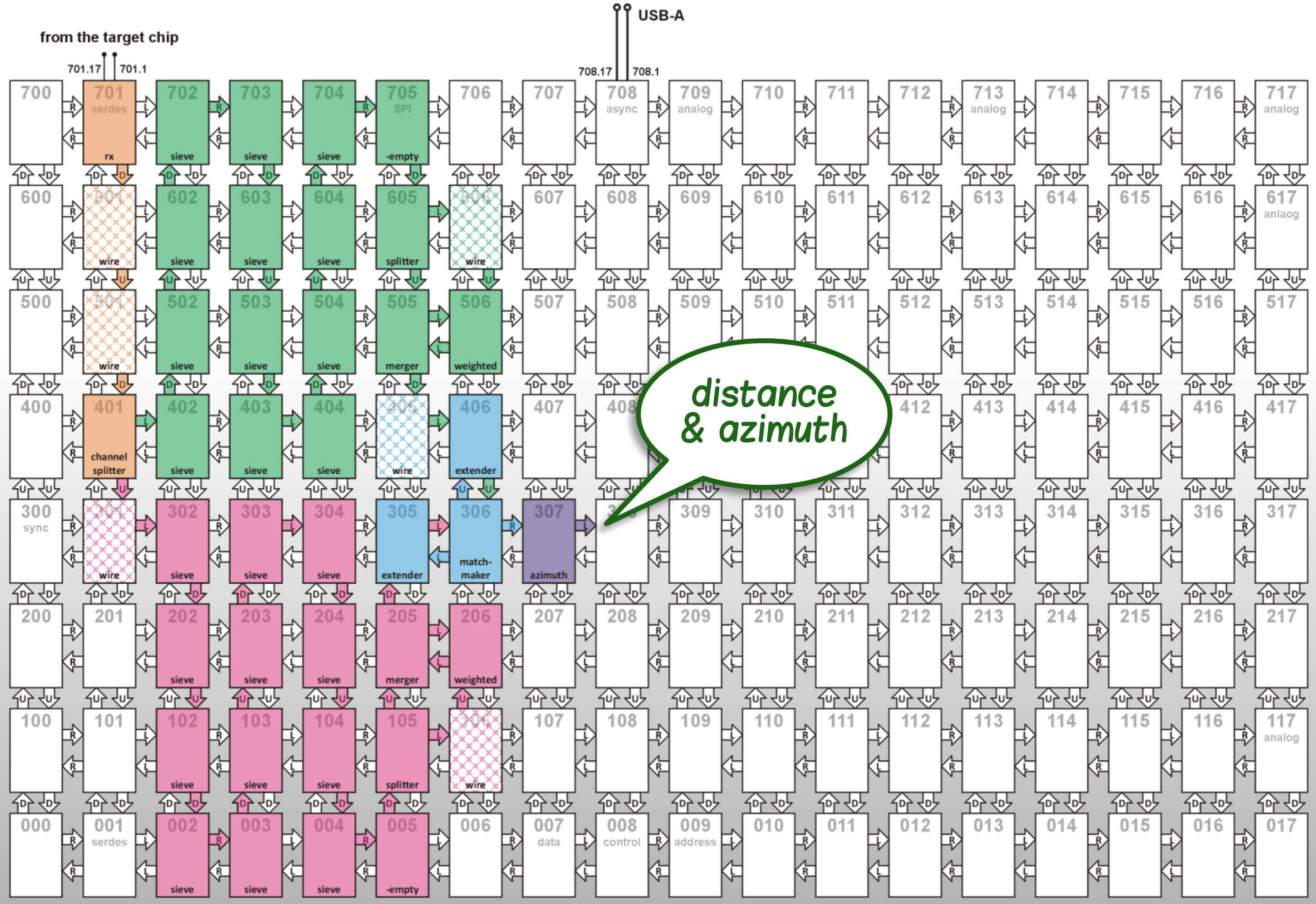
azimuth calculator




```
307 +node 307 /ram 1F5 r-l- /b 11 go /p

reclaim 307 node 0 org
abs -if neg - 1 . + then ;
sgn@ @p drop @p ; sgn! !p ; 0 ,
sgn sgn@ -if drop neg ; then drop ;
4/d a! +* +* a ;
/ hd-rq push dup dup or 4/d pop
neg --u/mod ;
go 11 @b -if !b go ;
then push @b @b over over . + dup push push
neg + dup sgn! abs pop / sgn
pop pop !b !b !b go ;
```

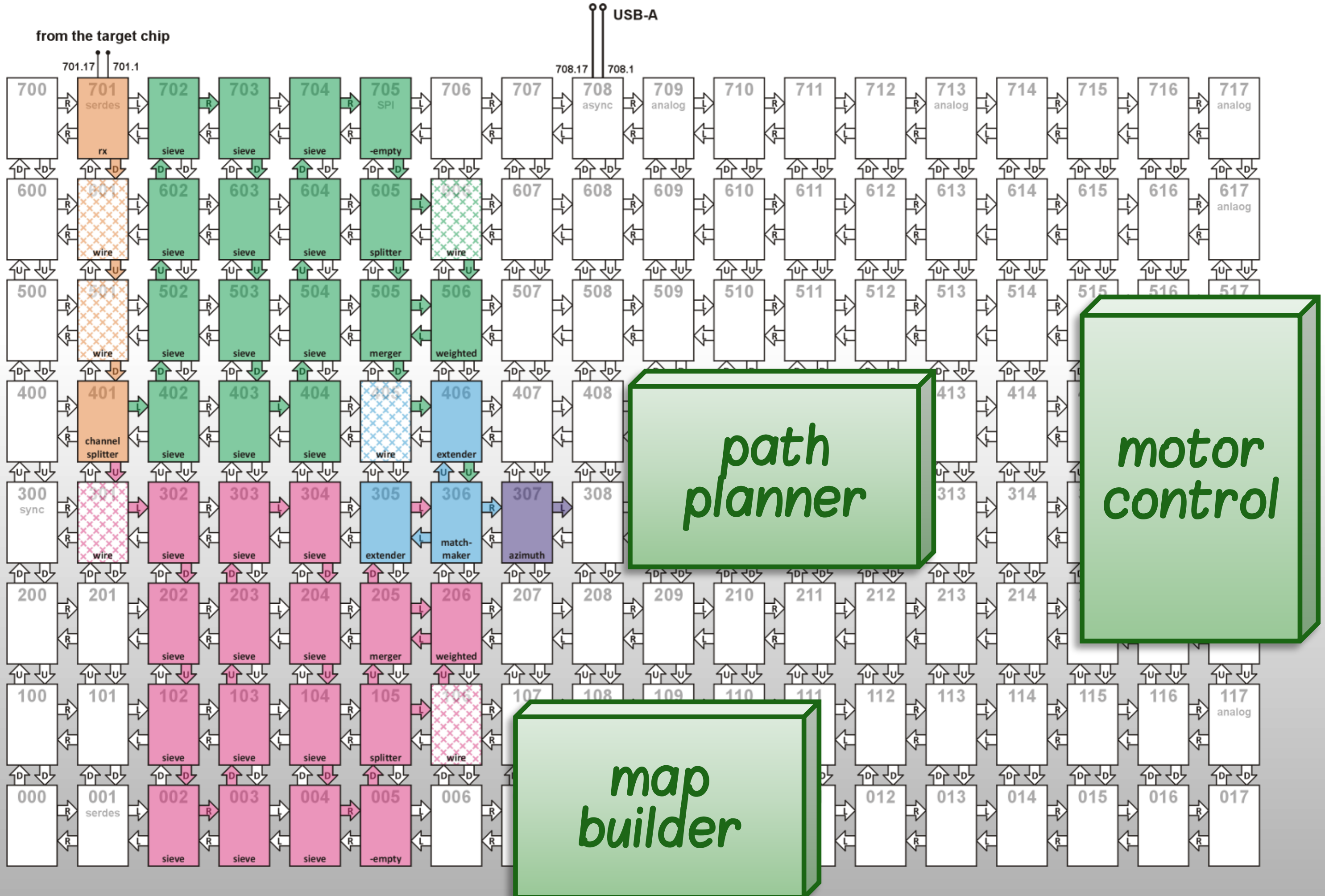
host chip

possible use of distance & azimuth



host chip

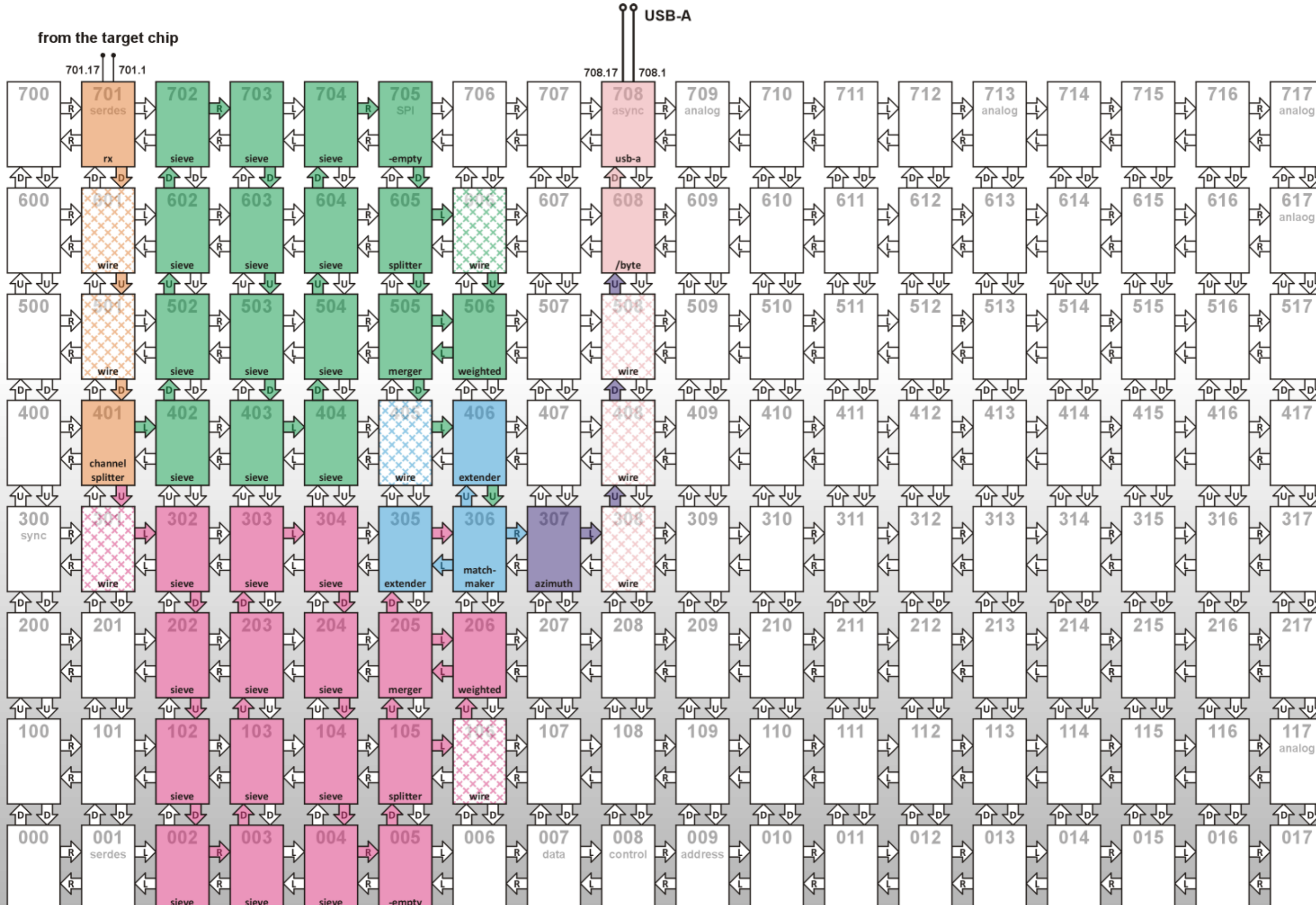
possible use of distance & azimuth



host chip

transfer to PC

distance
& azimuth



DEMO #3

„radar screen“

link to demo #3 video on YouTube

FURTHER DEVELOPMENT

possible improvements

- larger receptive field
  - wide angle ultrasonic transducers
  - adding pinnae (auricles)
- longer range
  - variable gain receiver amplifiers
  - more gmm calculators
  - stronger ping (more cycles)
- replace gaussians with other (asymmetric) functions

potential areas of interest

- detection of moving objects
  - Doppler shift
- vertical obstacle detection
  - frequency sweep ping
  - spectral analysis
- size, shape, surface roughness estimation

acknowledgements

Dr. Louise Allen and Dr. Nickolay Hristov
high-speed video of bats emerging from Carlsbad Caverns

Dr. Aaron Corcoran
high-speed video of a Myotis bat capturing insect

Michael Kuhlman
reproductions from Bat-inspired robot navigation technical report

Greg Bailey & GreenArrays, Inc.

Chuck Moore

contact information

Email: dkalny@seznam.cz   Skype: live:dkalny