

2-D PT module concept for the sLHC CMS tracker

Special thanks to
M. Pesaresi, M. Raymond, A. Rose
A. Marchioro
CMS Track-trigger task force
CMS Tracker upgrade simulation team

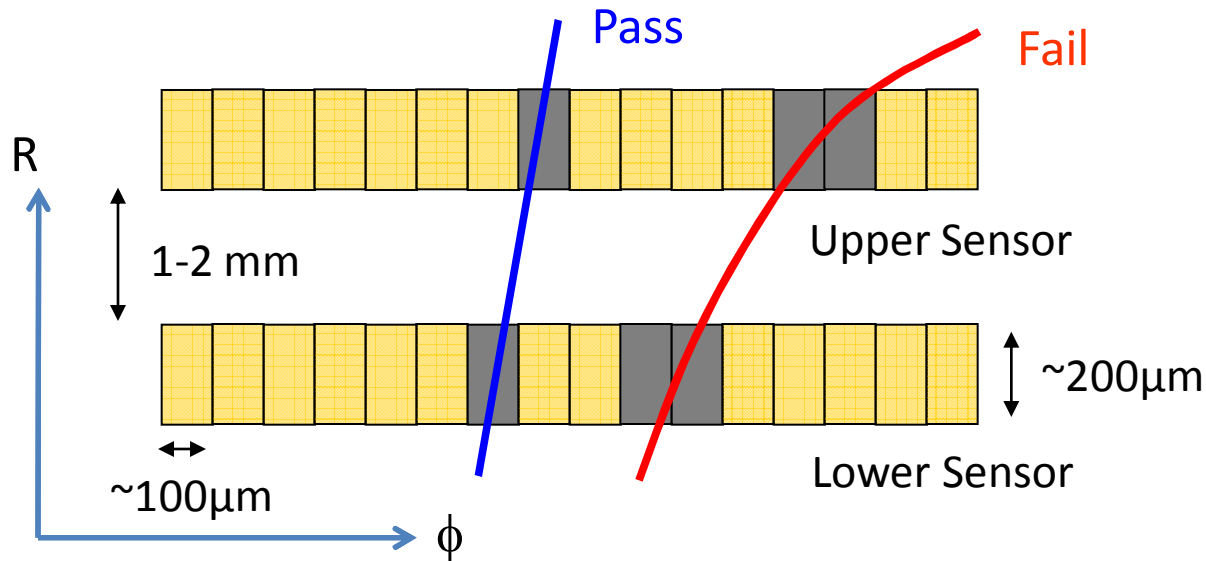
Geoff Hall

Outline

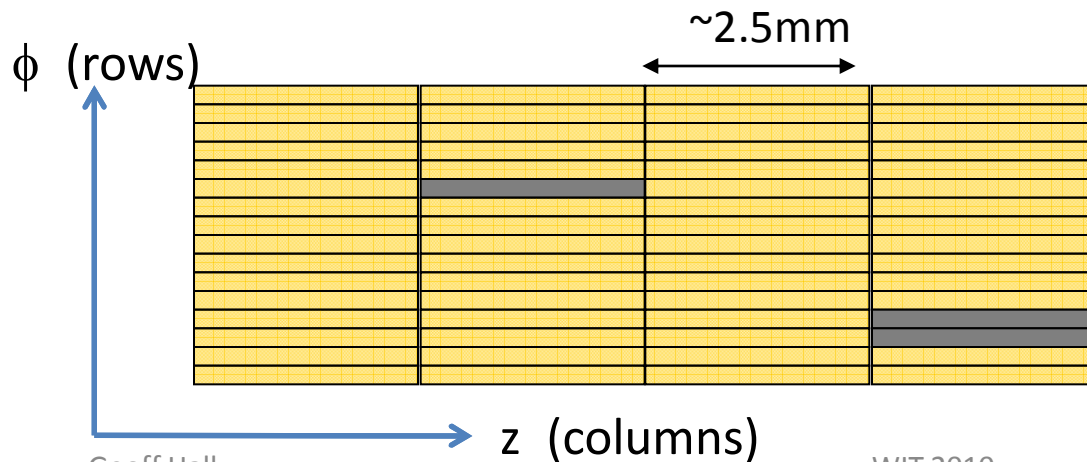
- Basic idea of, and motive for, PT module
 - already presented by M Pesaresi
- Describe one possible way of building such a module
 - not fully worked out
 - ideas are still evolving and we do not know the best route
- The requirements are not fully defined
 - originally the long term upgrade aimed to operate at $10^{35} \text{ cm}^{-2} \cdot \text{s}^{-1}$
 - it's obvious that this is a long way in the future and we should expect the requirements might evolve with the LHC physics discoveries and the operation of the LHC machine, which still faces many challenges
 - nevertheless, the LHC physics programme is of utmost importance and the longevity of the accelerator
 - hence R&D on new module concepts will prove invaluable
 - crucial to advance the means to construct advanced module types
 - must understand better the technical drivers to assemble these modules since significant investment is required

Basic module requirements

- Compare binary pattern of hit pixels on upper and lower sensors



High p_T tracks can be identified if hits lie within a search window in $R-\phi$ (rows) in second layer



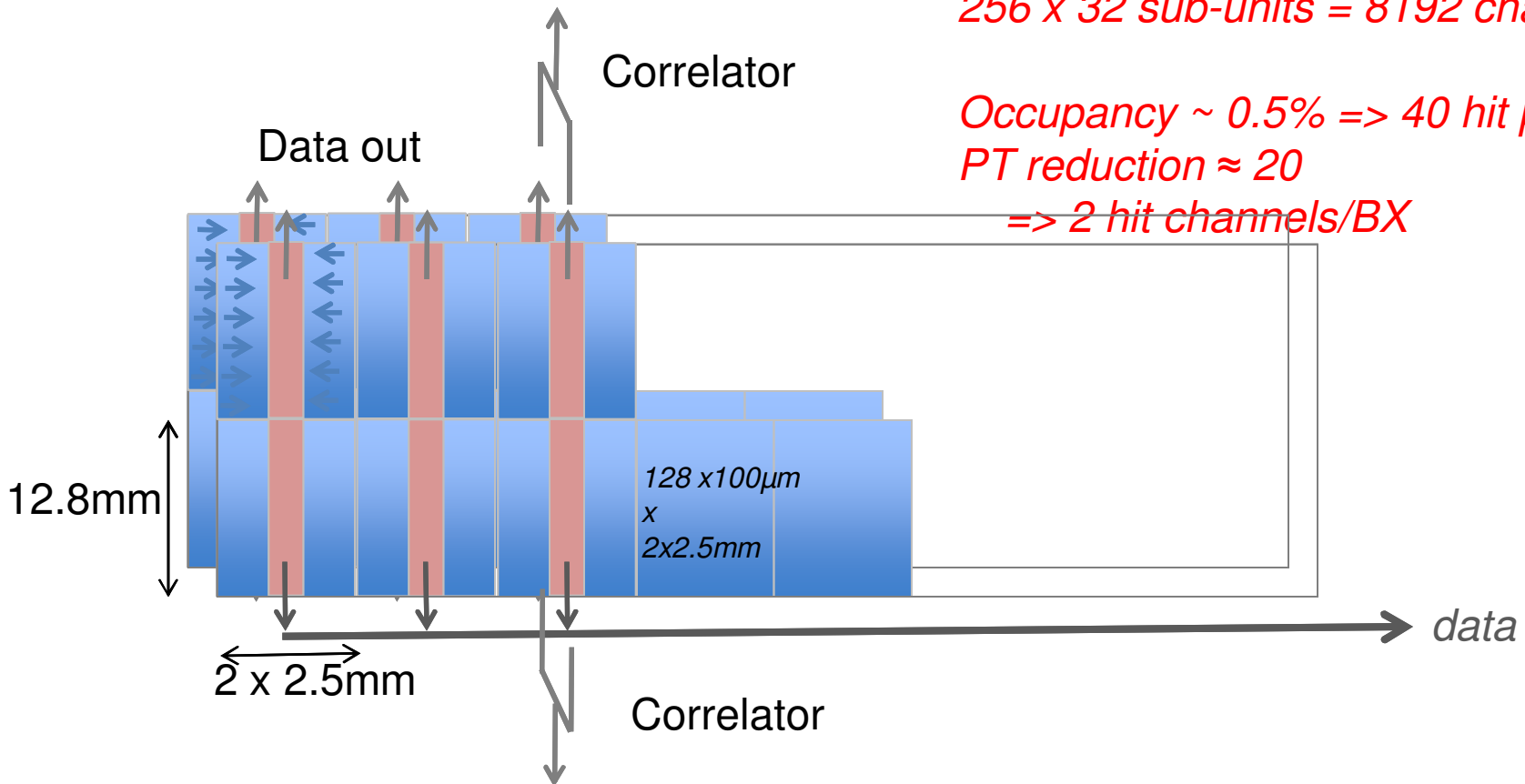
Sensor separation and search window determines p_T cut

Schematic of PT module

- Transfer hits to both edges –with minimal power – for comparison logic
 - also store hits on pixel for L1 readout

*Module 25.6mm x 80mm
256 x 32 sub-units = 8192 channels*

*Occupancy ~ 0.5% => 40 hit pixels
PT reduction ~ 20
=> 2 hit channels/BX*



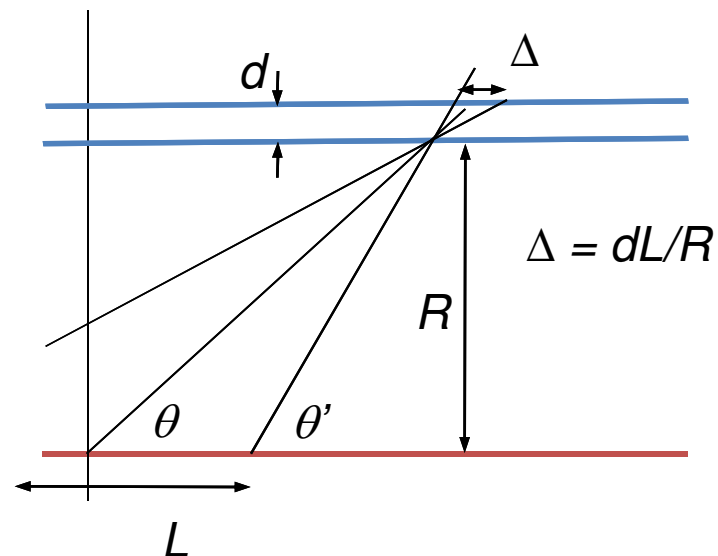
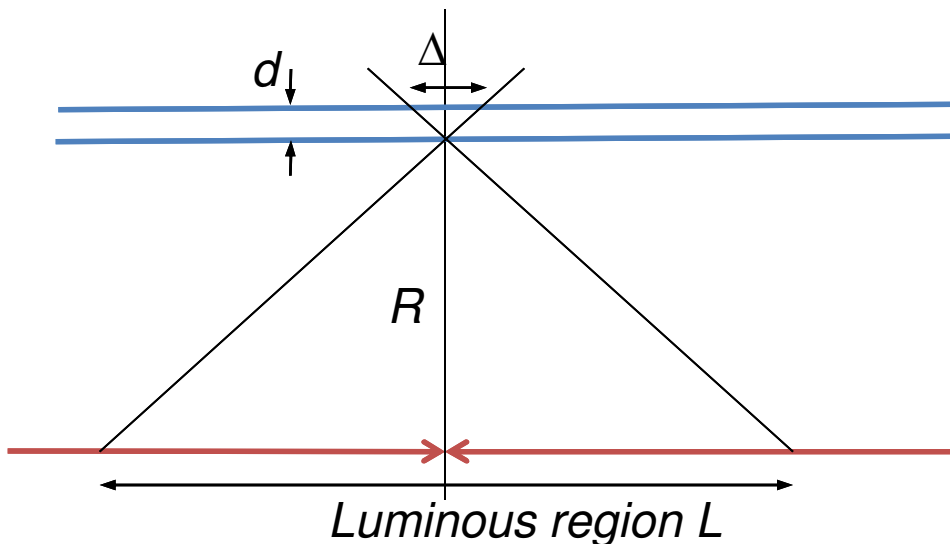
What defines module size?

- Assumed radial location $\sim 25\text{-}45\text{cm}$
 - allows to cover full CMS η range, with only barrel assembly
- Pixel size $\sim 100\mu\text{m} \times 2.5\text{mm}$
 - $\sim 100\mu\text{m}$ – matches required resolution and likely assembly precision for double layers
 - $\sim 2.5\text{mm}$ - defined by approximate projection of luminous region at $R \sim 25\text{cm}$ and likely radial spacing $\sim 2\text{mm}$ (next slide)
 - coarse enough for low cost bump bonding (or perhaps even wire for prototyping)
- 256×32 - binary multiple of pixels with practical dimensions
- $\sim 25.6\text{mm} \times 80\text{mm} \Rightarrow 4$ sensors in 150mm wafer – ie one contribution to yield
- Chip height $\sim 15\text{mm}$ – fits reticle
- Expected occupancy at $R = 25\text{cm} \sim 0.5\% @ 10^{35}$
 - so average < 1 hit in column which may allow transfer to edge
- Of course, this is history and explains initial parameters chosen for simulations
 - module can probably be enlarged, eg 384×32
 - but concept has avoided chip to chip connections – i.e. > 2 ASICs in height

PT layer pixel size

- R- ϕ : compare to likely assembly precision $\sim 100\mu\text{m}$
- Should reduce need to compare many nearby columns
 - Δ independent of η , but offset in z between layers increases with η

$R = 25\text{cm}$, $L = 28\text{cm}$ $d = 2\text{mm}$ $\Delta \approx 2\text{mm}$



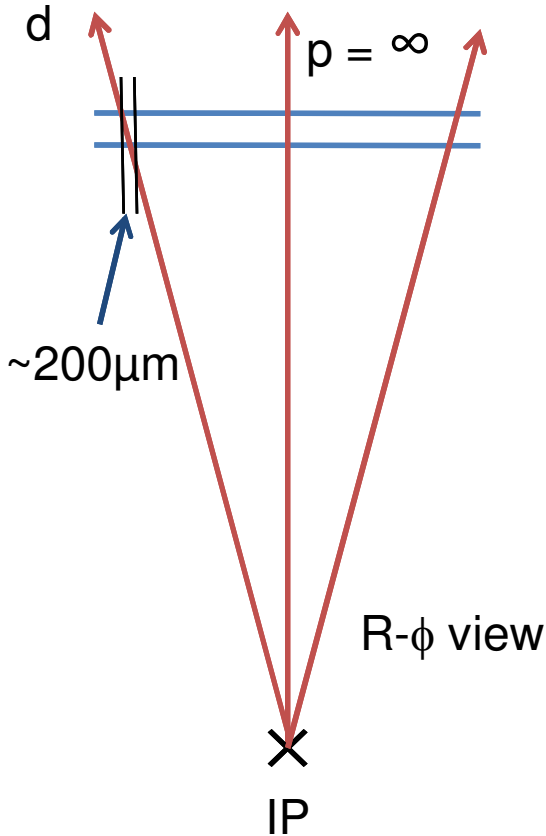
LHC luminous region $L \approx 28\text{cm}$ ($\pm 3\sigma$) – may be larger or smaller at SLHC

Why edge-readout?

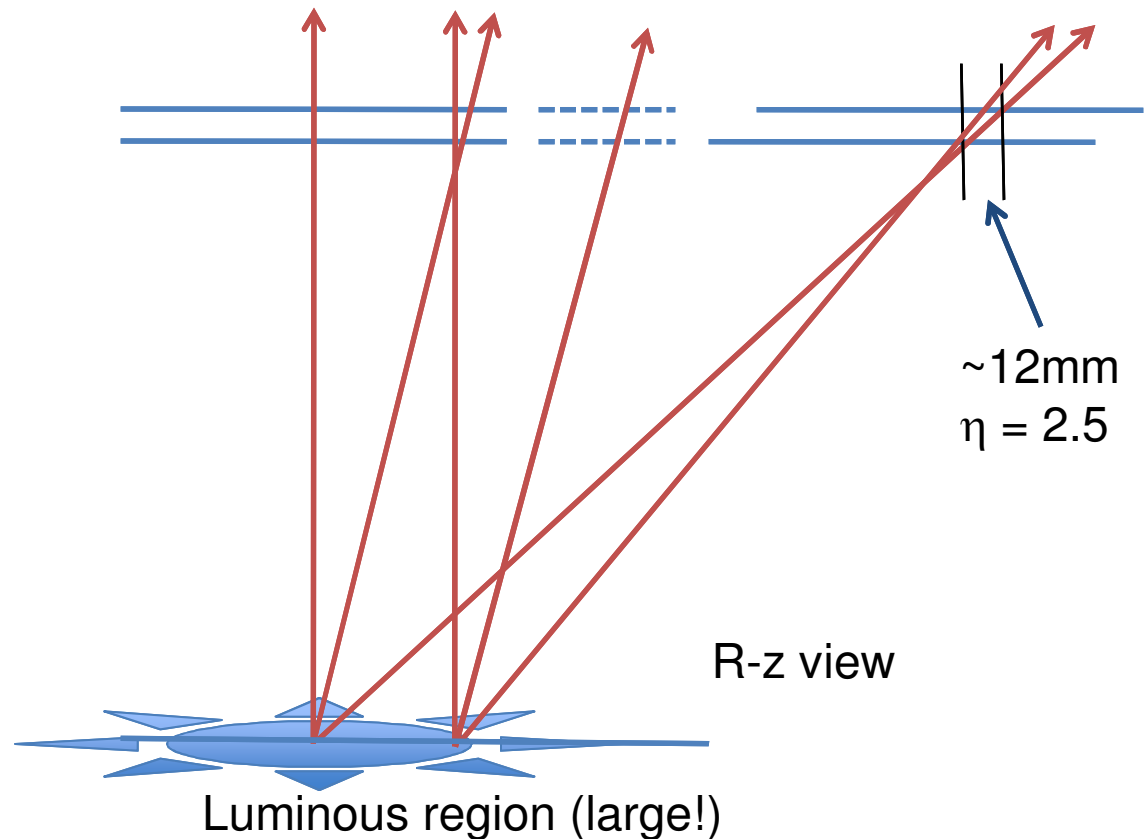
- Original motivations were:
 - it was (is) far from clear how to solve layer to layer interconnection problem
 - edge readout seemed to offer means to factorise this, eg by constructing a small, dedicated component
 - real connectors close to requirements do exist but would probably not be usable for mechanical reasons for ~8000 connections
 - profit from low occupancy and high density of lines on ASIC
 - no penalty in making comparisons at edge of module
 - constructing two module layers independently, then making final assembly is attractive conceptually
- This logic exposes some prejudices (which could be wrong)
 - module should evolve from known technologies
 - design should profit from known constraints (ie occupancy)
 - design should be adaptable to changes in requirements
 - a unique double layer assembly - which is best commercially – may not be possible
 - the logic to be included is not yet well defined and may also evolve

Comparison logic

- Modules are flat, not arcs
 - Compensate for Lorentz drift
 - Orientation of module
- => position dependent logic



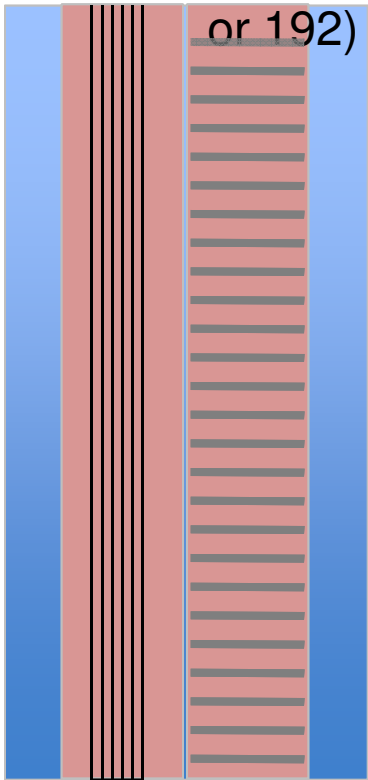
- z offset η dependent
- search window to allow for luminous region and quantization => 3 pixels (if not tiny)



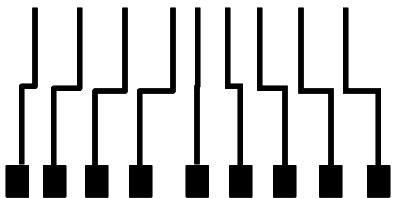
- Family of modules with offsets in z

Possible PT layer readout

column of 128 (or 192) pixels

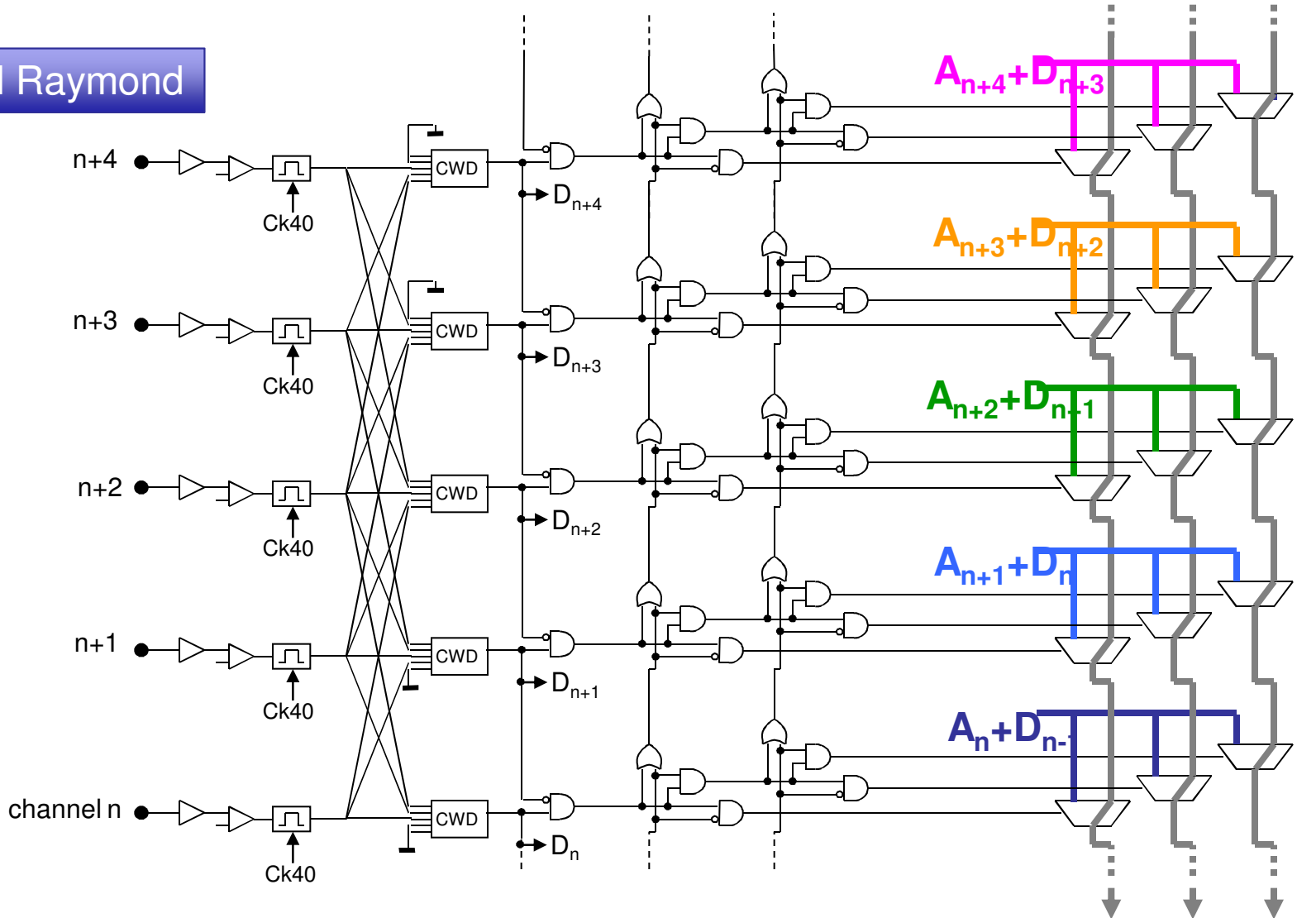


- shift register ruled out: $128/25\text{ns} = 5.12\text{Gb/s}$
 - probably even if allow several BX and latency penalty
- worst case occupancy may mean >1 hit/BX
 - with adequate fluctuations to be permitted
 - NB from simulations, jets don't have much impact
- equip column with $N \times 8$ address/data lines
 - N = maximum number of clusters allowed
 - ignore combinations consistent with wide clusters
 - seems easy to read out 3, or more clusters (also suspect occupancy is pessimistic)



one possible design for transferring up to 3 cluster addresses to end of column logic

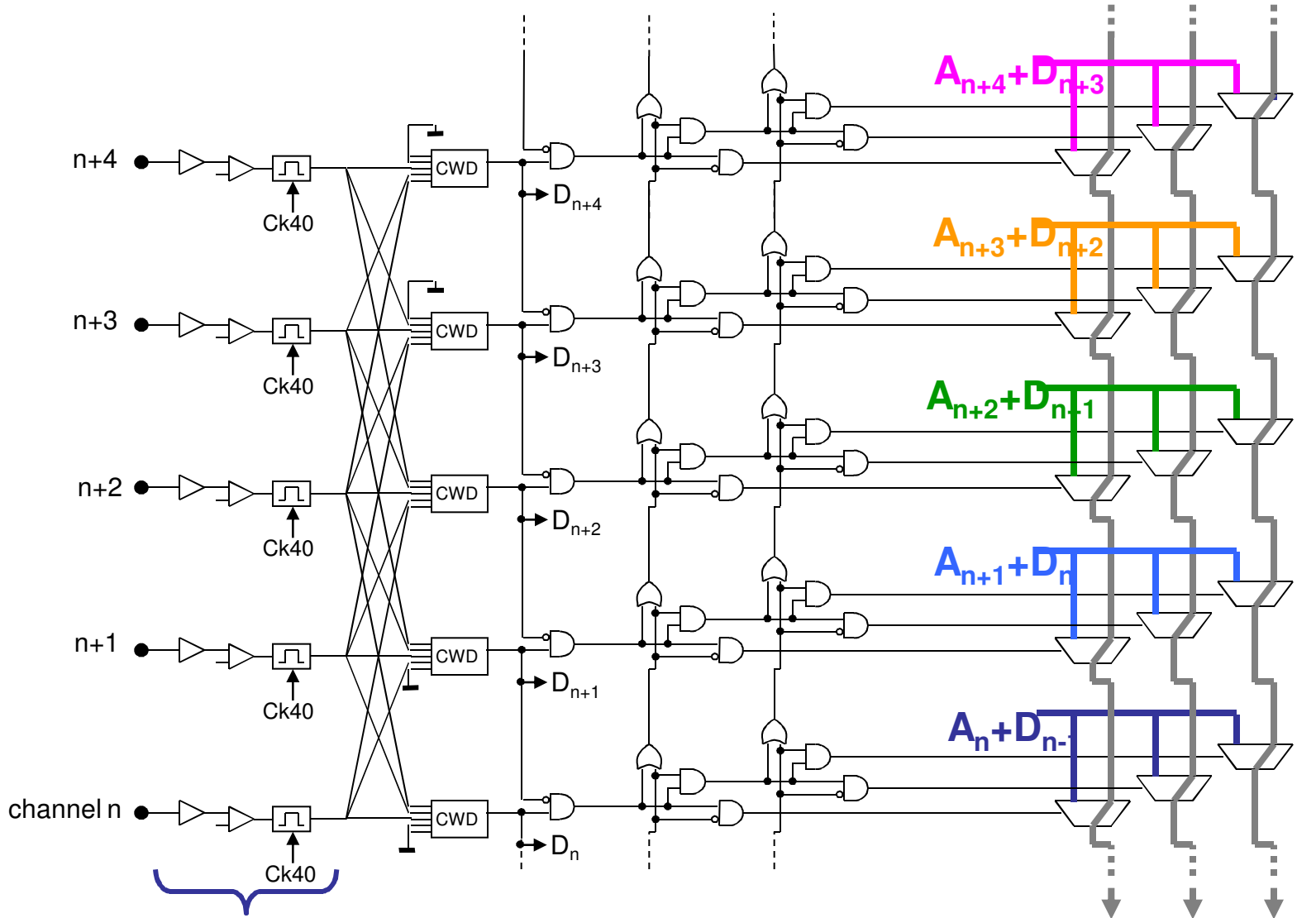
M Raymond



go through logic functionality stage-by-stage
then show some examples

end of column
logic

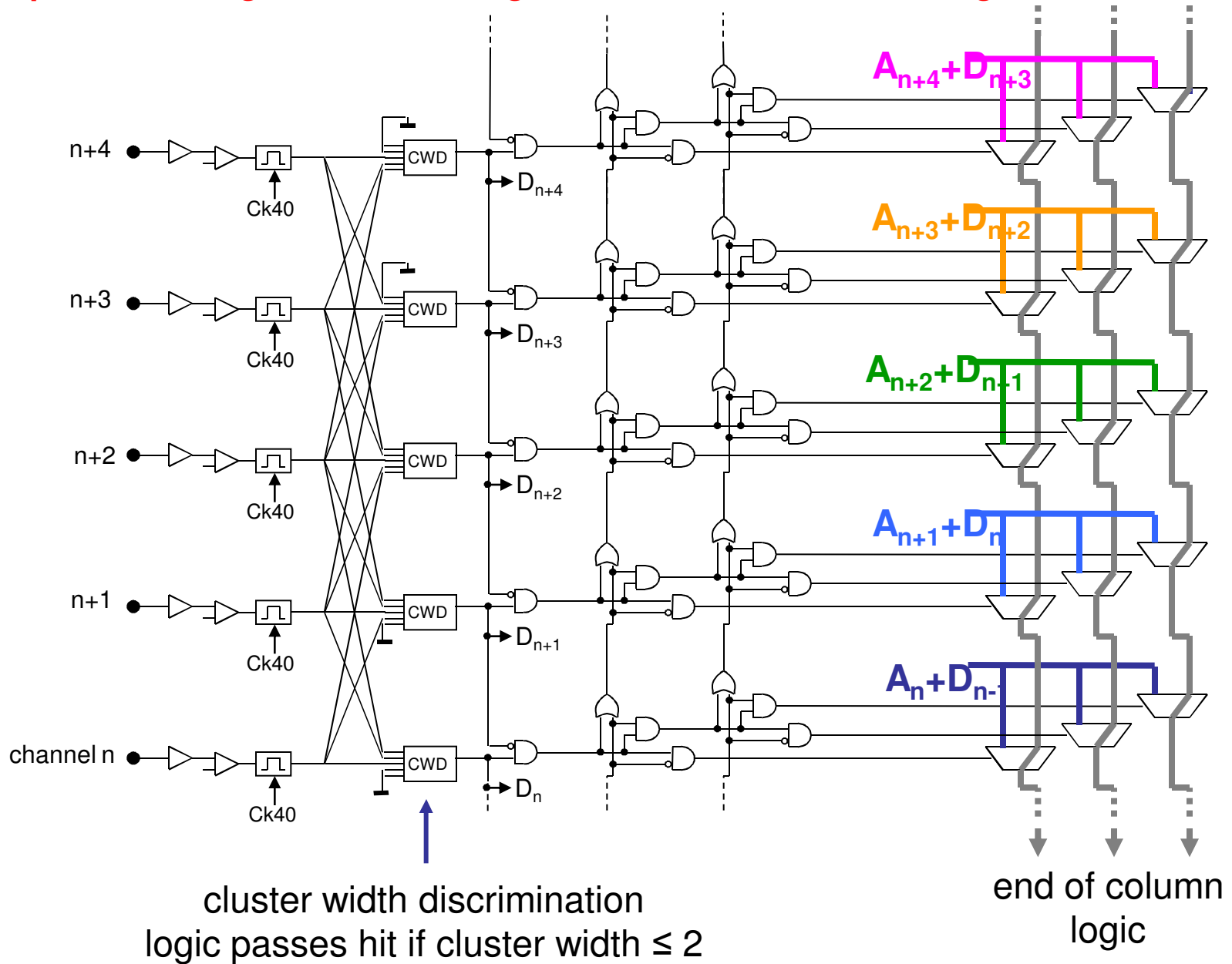
one possible design for transferring up to 3 cluster addresses to end of column logic



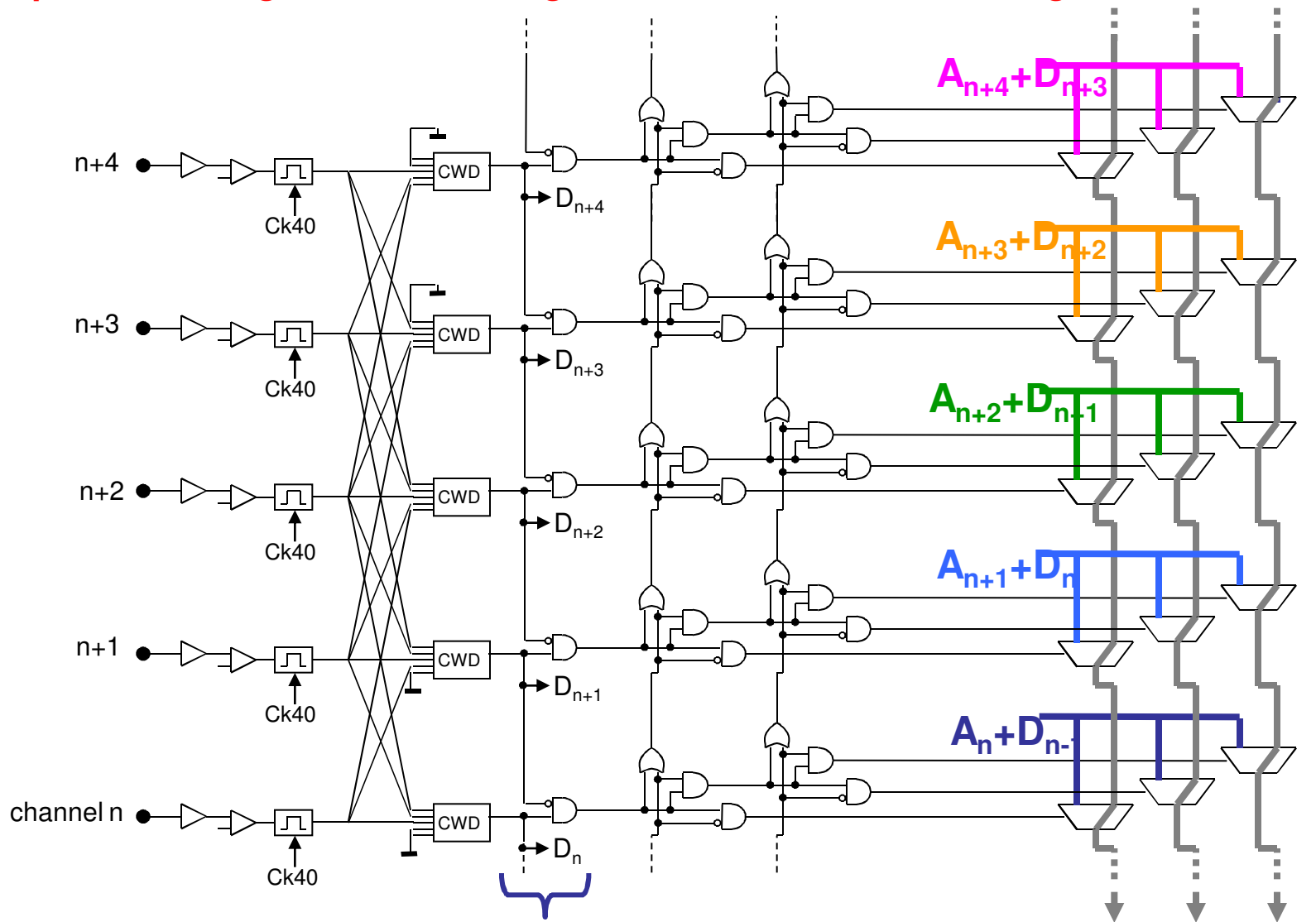
front end produces a 25 ns pulse
if signal exceeds comp. thresh

end of column
logic

one possible design for transferring clusters to end of column logic



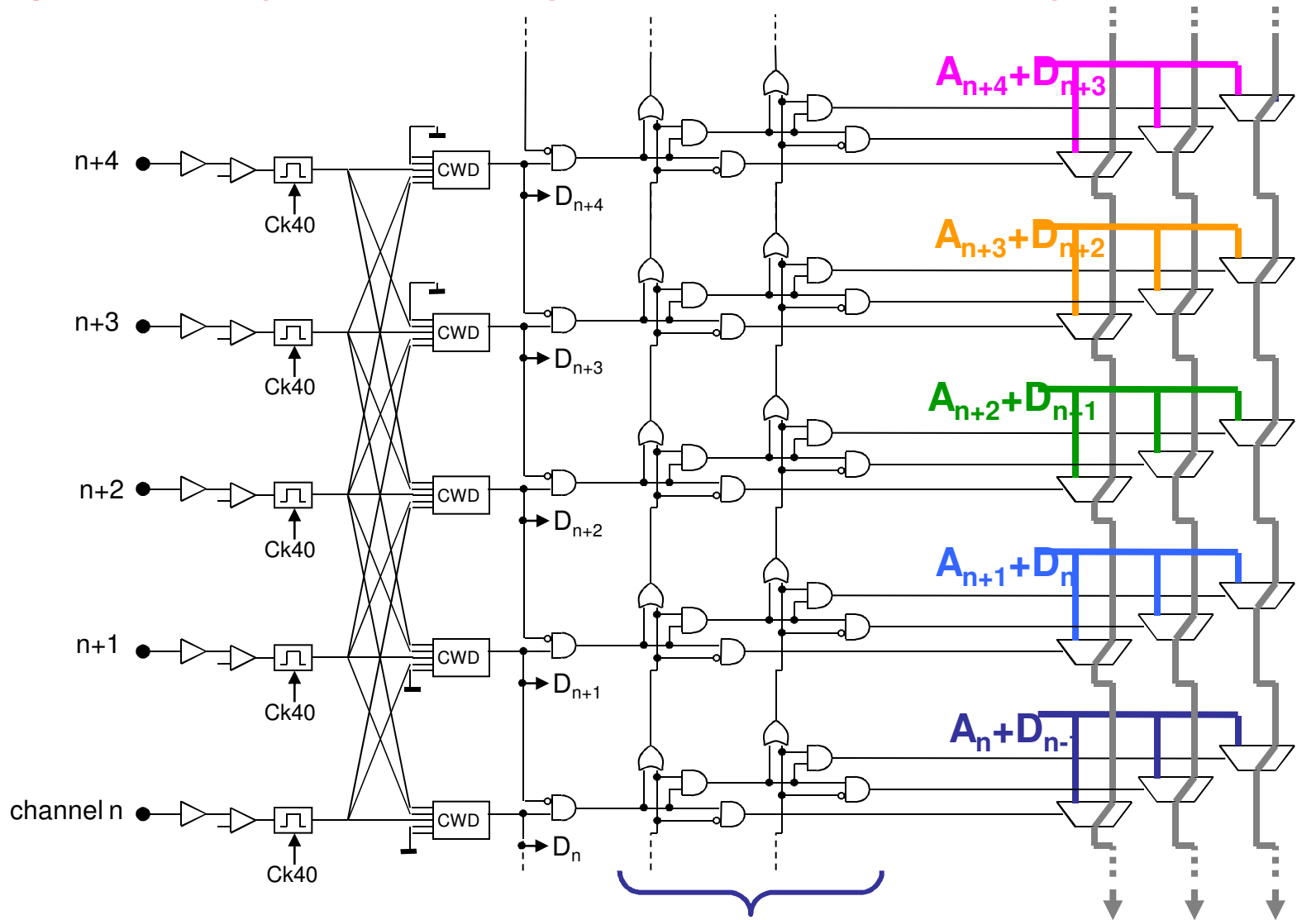
one possible design for transferring clusters to end of column logic



further logic ensures that only one channel signals a hit even if it is part of a two channel cluster

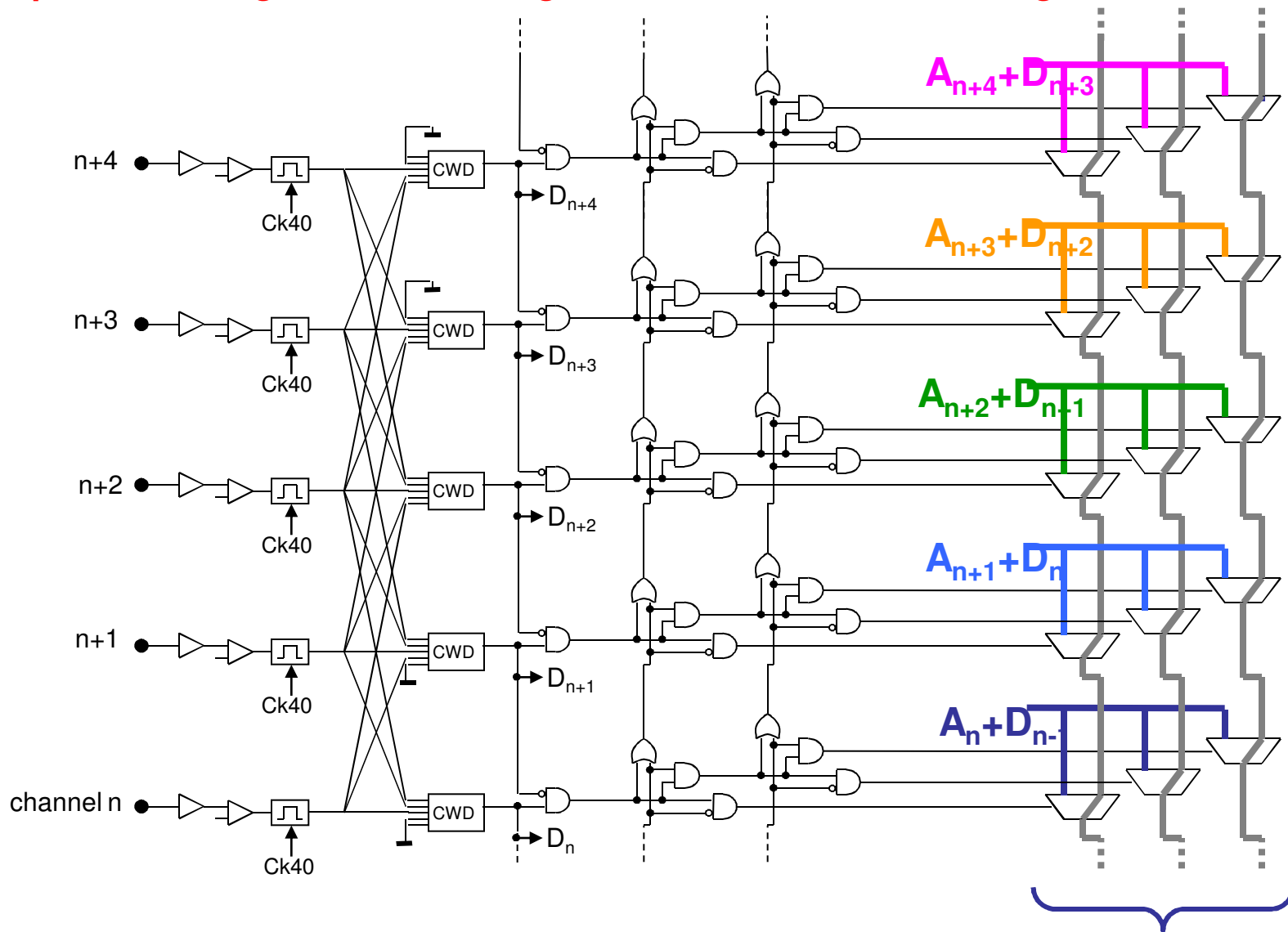
end of column logic

one possible design for transferring clusters to end of column logic



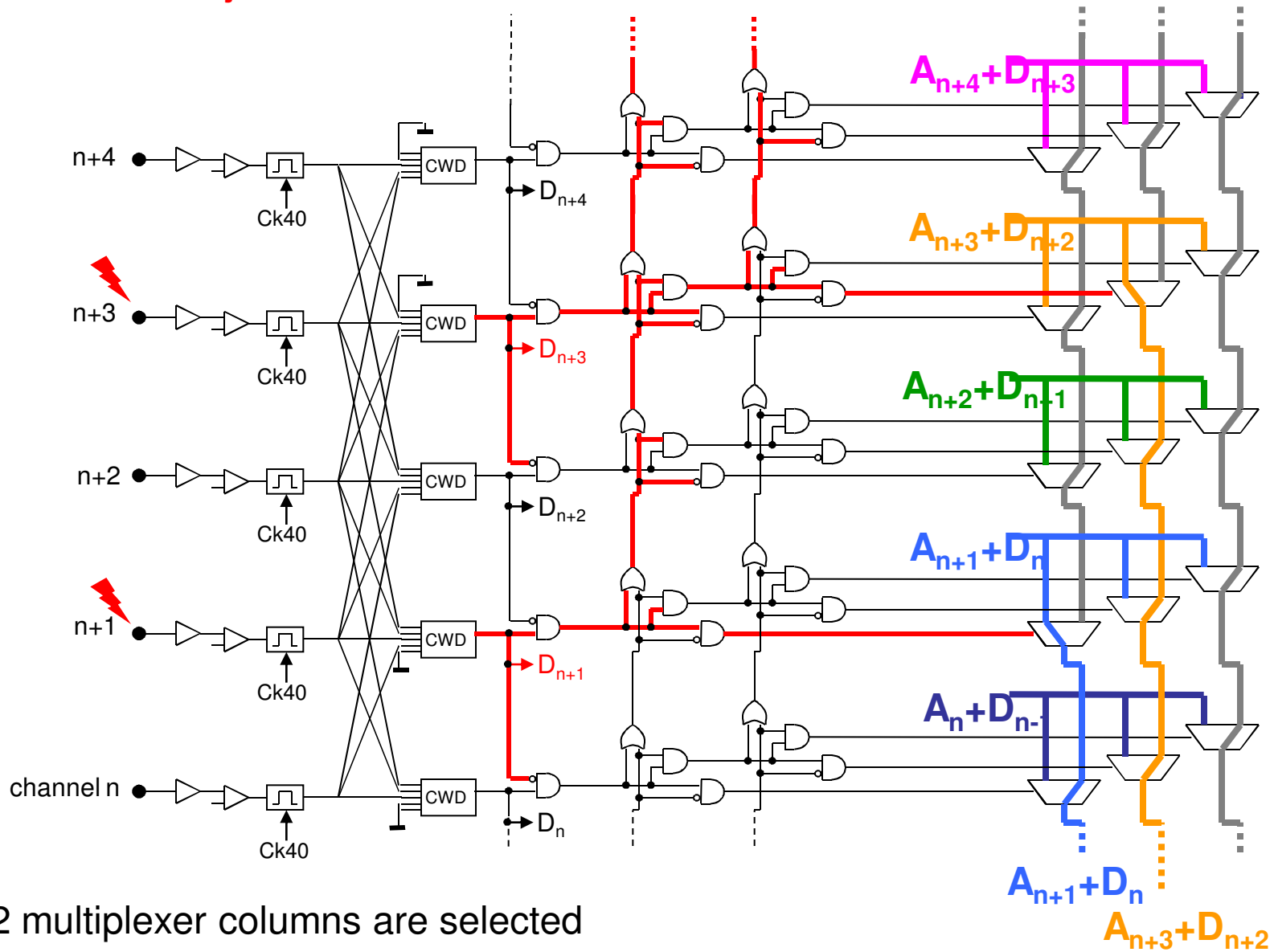
priority logic to determine which end of column
mux is active for a particular cluster logic

one possible design for transferring clusters to end of column logic



up to 3 cluster addresses can be transmitted to end-of-column logic in this example

2 hits on non-adjacent channels

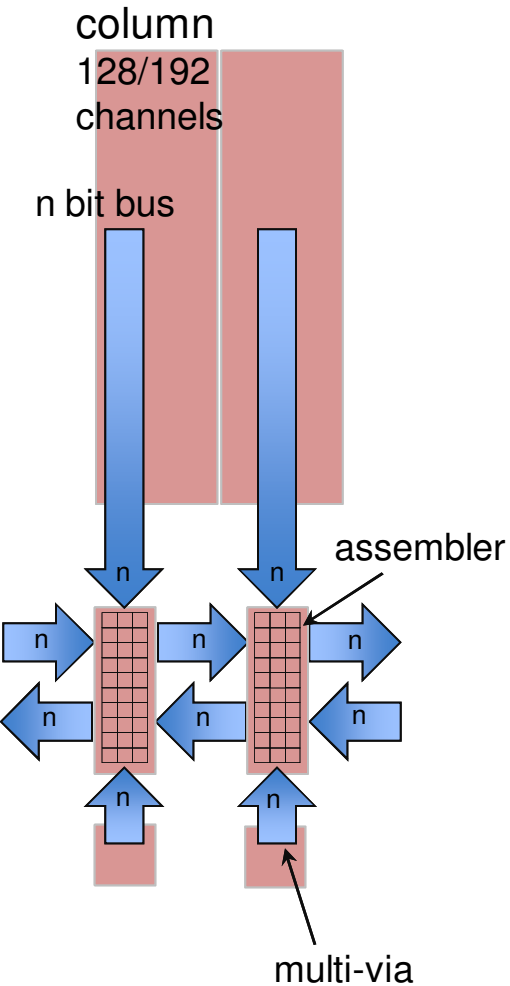


2 multiplexer columns are selected

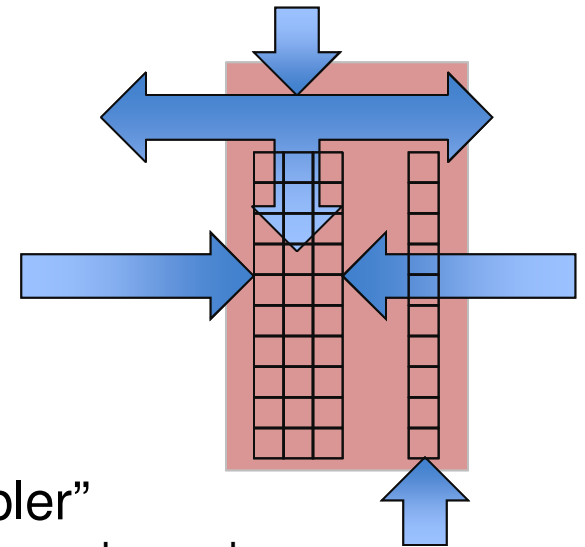
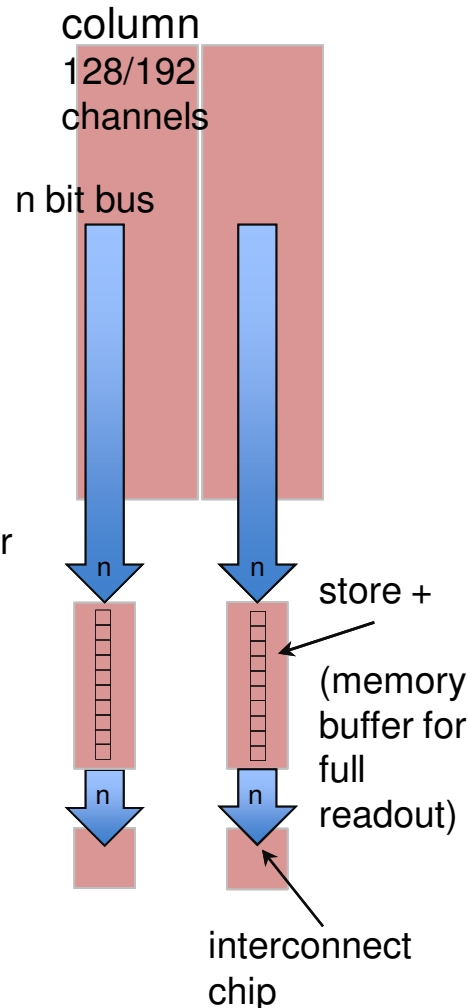
other examples in backup slides

Track stub generation by matching layers

upper layer



lower layer



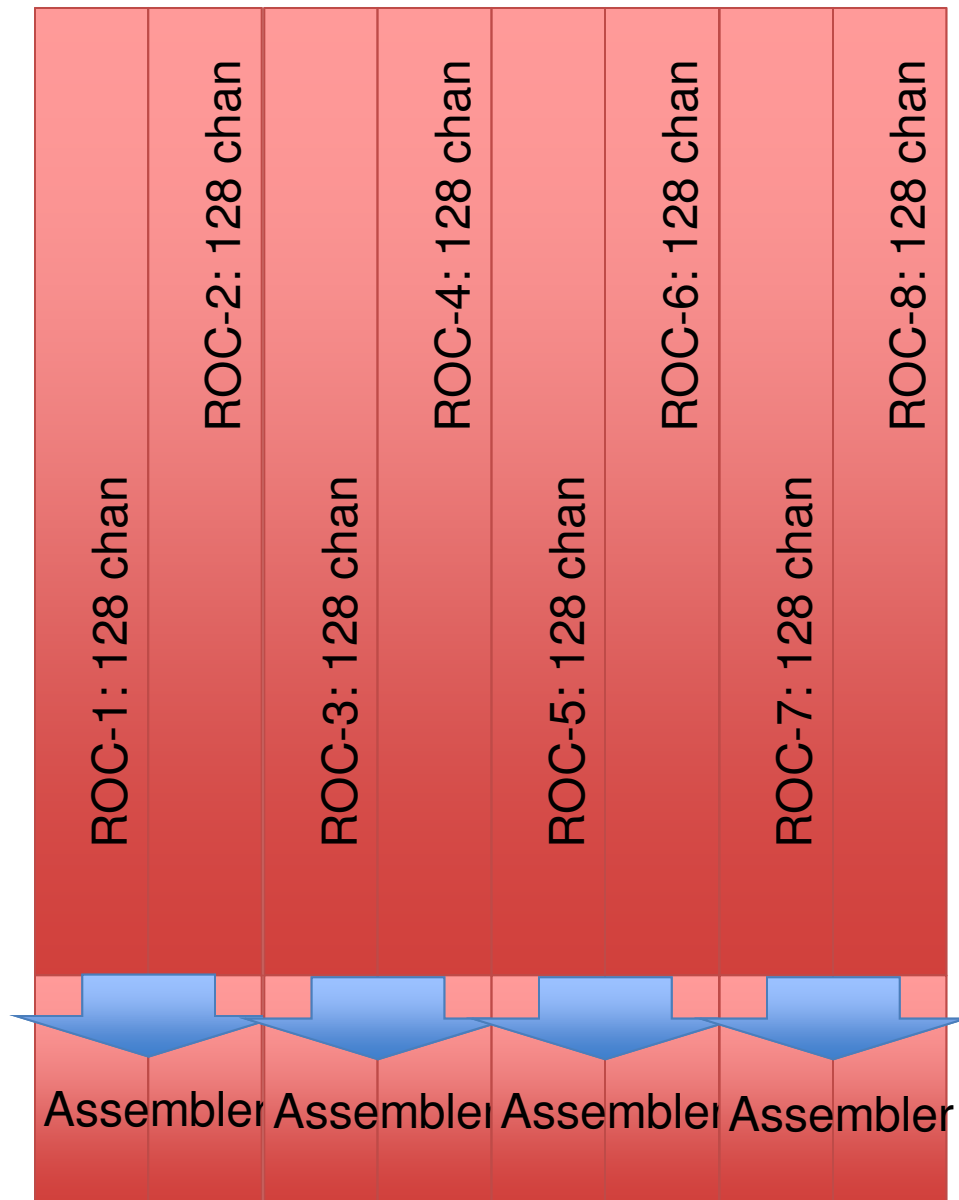
“assembler”

n bits from column above
transmits column to each neighbour
receives n inputs from each neighbour
and stores
receives n inputs from module below

compares pattern from module below with
three (n bit) stored patterns

need to proceed to VHDL simulation

Layout



Make ROC + Assembler as single ASIC
identical for both layers
may switch off elements

Practical to produce a chip to serve several
columns which eases data transfer and
line density

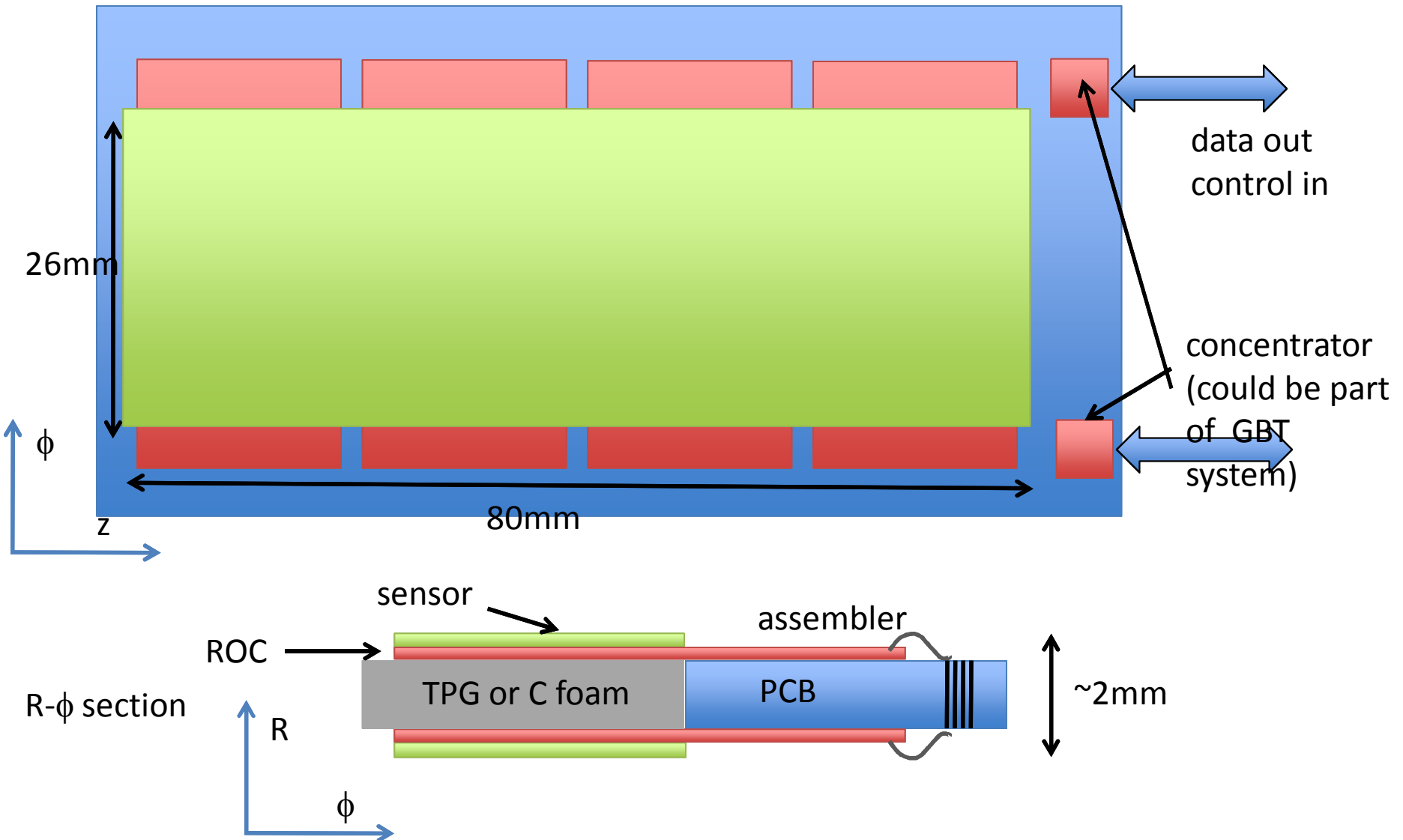
Looks easy to match 4 x 2.5mm columns
eg 2mm pitch

Looks possible to cover 8 columns with
pitch \approx 2mm

Width: 16mm to read 8 x 2.5mm

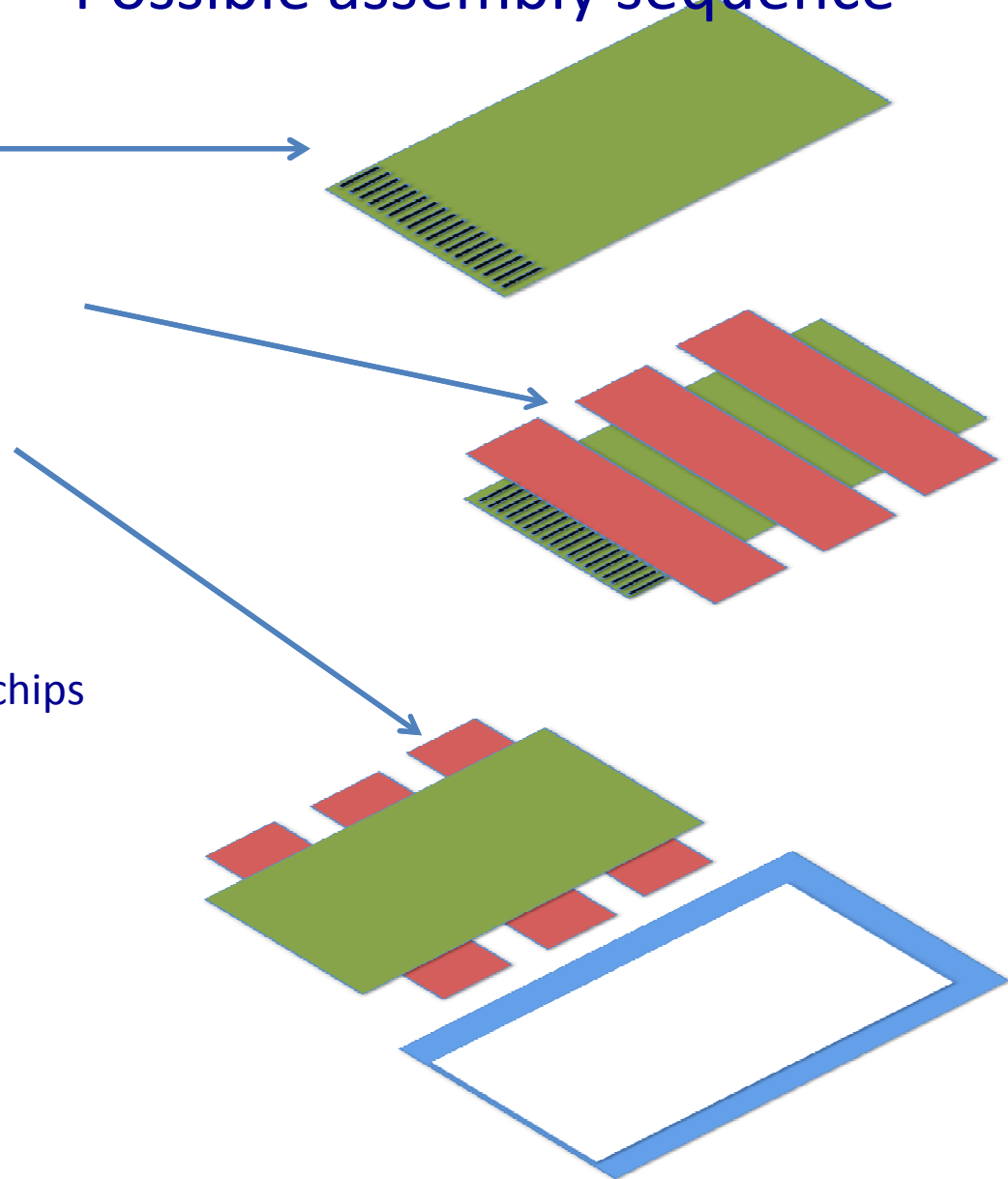
Chip size \approx 18mm x 16mm (128)
 \approx 21mm x 16mm (192)

Module schematic



Possible assembly sequence

- Sensor
- ROCs bump bonded to sensor
- Invert sensor-ROC object
 - Exposed ROC areas then face up
- Place assembly on hybrid
 - hybrid has pre-mounted ancillary chips
 - Wire bond ROCs to hybrid
- Prepare partner module
 - assemble and connect together



Weak points in this approach

- It is worth emphasising that this concept was originally to kick start a design effort. Nevertheless, so far insufficient attention given (by us) to several important issues
 - cooling – seems feasible to include thermally conductive layers
 - but not necessarily more difficult, as no interior interlayer connections
 - location of links – most likely to be on, or close to, the module
 - originally it seemed that links could be deployed on a remote bulkhead
 - this now seems implausible – high speed electrical links are undesirable for noise and material reasons and there are no obvious close locations
 - optimal matching of bandwidth and power (if adjustable) not yet obvious
 - provision of power to module
 - we assume DC-DC conversion and local converters
 - overall mechanical design
 - involves constraints above

Data volumes and link requirements

- Assume 24 bits/hit to transfer in each 25ns BX
 - includes time stamp and error coding

send trigger data from **one** layer of stack

for 40M channels in stacked layer

$$L = 10^{35} \text{ cm}^{-2}\text{s}^{-1}$$

Channels/chip	128
Occupancy	0.005
PT data reduction	0.050
Channels above PT cut/BX/layer	5,000
bits/channel	24
No links @ 5Gbps (3.2Gbps data)	1,500
Power/link [W]	2.0
Link Power [kW]	3.0
Power/chan [μ W] with 50% BW usage	150

Comments

150 μ W is conservative estimate for trigger data only

assumes 50% use of bandwidth
and 2W 5Gbps GBT

GBT power may improve
ideally should optimise power & speed

but **additional links required** for full readout

with 6.4 μ s storage on each FE pixel

Power estimate for PT module

	P [μ W] per pixel	Functions
Front end	25	amplifier, discriminator local logic, cf ATLAS 130nm pixel
Control, PLL	10	1 PLL/ROC @ 5mW, x 2
Digital logic	2	transfer to edge (M Raymond)
Comparison	4	logic (0.5mW/column)
Data transfer	0.25	few cm across module
Data to local GBT	0.25	transmit 48bits/BX @ 1pJ/bit? \approx 2mW/module
Concentrator	5	buffer to and from GBT: 2 ASICs @ 20mW
Full readout	20	following L1 trigger, extrapolate from CMS pixel
Sub-total	\sim 67	
Total with DC-DC	\sim90μW	75% efficiency for DC-DC conversion

NB big uncertainties and significant guesswork

eg SEU-robustness, full control and timing, data volumes,....all required

essential to improve on this with real design work

Other considerations

- “Simplicity” is desirable
 - designers and users may have different perspectives on issues such as
 - grounding and shielding
 - control software development
 - initialisation and data unpacking software
 - off-detector firmware and digital processing (trigger system will probably continue to be debugged with real triggers only in-situ)
- Up to now, there was modest overlap between real end-users and ASIC designers
 - this will probably be more crucial for these modules
 - substantial evaluation programme should be foreseen

An parable of evolution

- To make soap powder, liquid is blown through a nozzle.
- As it streams out, the pressure drops and a cloud of particles forms... thirty years ago, the spray came through a simple pipe that narrowed from one end to the other... it had problems with irregularities in size of grains, liquid or blockages...
- the nozzle has become an intricate duct, longer than before, with many constrictions and chambers. The liquid follows a complex path before it sprays. Each type of powder has its own nozzle design which does the job with great efficiency.
- The problem was too hard to allow even the finest engineers to explore with mathematics and design... they tried another approach... **evolution**: preservation of favourable variations and rejection of those injurious.
- Take a nozzle that works quite well and make copies, each changed at random. Test them for how well they make powder. Then impose a struggle for existence by insisting that not all can survive.
- Many altered devices are no better (or worse) than the parental form. They are discarded, but the few able to do a superior job are allowed to reproduce and are copied – but again not perfectly. As generations pass, there emerges a new and efficient pipe of complex and unexpected shape.
- from Steve Jones. *Almost Like a Whale*.

Conclusions

- The requirements for trigger modules are not yet clear
 - especially the physics case and luminosity scenario
 - a lot of factors involved in constructing the module
- Power is crucial for such layers in future trackers
 - the trigger data transfer still dominates but maybe can be optimised
- Evolution is about finding a solution which matches the environment
 - intelligence may not be what we think it should be
 - we need to try out alternatives and really evaluate them
- It may be necessary to think hard about using expensive technologies in a way which allows us to try variants before committing to a single solution

Backups

Approximate parameters of trigger layers

For stacked layer (doublet)	
Pixel size	100 μ m x 2.5mm
ROC	8 x 128 channels
<Power>/pixel	250 μ W (*)
$ \eta_{MAX} $	2.5
Bandwidth efficiency	50%

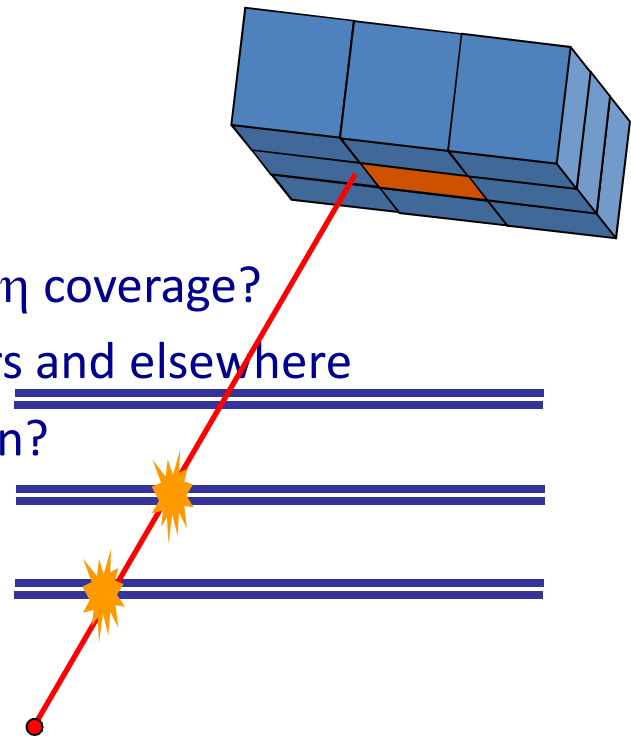
R [cm]	L [m]	A [m ²]	N _{face}	N _{chan}	N _{ROC}	N _{module}	N _{links}	P [kw]
25	3.0	9.6	64	38.5M	38k	4700	2880	9.6
35	4.2	18.7	88	75M	73k	9200	5610	18.7

(*) With overlaps in R- ϕ or η expect additional 10-15%

present tracker ~35kW

Making a trigger

- Stubs provide track trigger primitives
- Not yet proven how these contribute to trigger
 - and rate reduction achievable
 - many simulation studies under way
 - expect to match a series of stubs to a calorimeter or muon object
 - using off-detector processors
- Questions to answer include
 - how many layers are needed?
 - what is the optimal location, allowing sufficient η coverage?
 - what is the impact of material? – in trigger layers and elsewhere
 - how important is z-measurement, and resolution?
 - what is the impact on tracking performance?
 - cost, power and material budget?
 - L0 trigger to guide?



8 bits data transmission through mux

hit occupancy per strixel column = $0.5\% \times 128 = 64\%$

93% of time only one cluster per 128 strixel column (simulation) => close to 100%

each mux line can change state 50 % of time (either '1' or '0', so 50% of time will change from

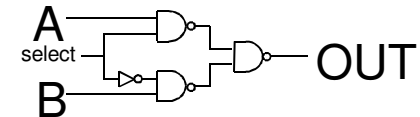
so translating to an “average” toggling speed

20 MHz (line can only change state every 25 ns)

x 0.64

x 0.5

= **6.4 MHz**



average power consumed in the mux gates

$2 \text{ (gates / mux line)} \times 8 \text{ (lines)} \times 6.4 \text{ (MHz)} \times 9 \text{ (nW / MHz / gate)} = 1 \text{ uW}$

(note: pessimistic since not all mux gates will be active – depends on location of hit)

= **~ 50 nW per strixel channel** negligible

transmission power associated with transmitting CMOS levels across chip

(note this power also consumed in the gate driving the line, but consider separate here)

(use 2 uW/MHz/cm)

$2 \times 6.4 \text{ (MHz)} \times 1.28 \text{ (cm)} \times 8 \text{ (mux lines)} / 128 \text{ strixels}$

= **~ 1 uW per pixel**

some digital power estimates (2)

40 MHz clock distribution: 100 μ W

(2 μ W/MHz/cm) (128 channel chip, 1.28 cm high)

= \sim 1 μ W / channel (assumes 1 V transitions, 2 pF / cm)

channel logic (including mux select):

\sim 30 gates toggling at channel occupancy frequency (0.5% x 40 MHz)

= 48 nW / channel (9 nW / MHz / gate) **negligible**

=> digital total associated with correlation data to edge of chip only \sim few μ W / channel

some circuit area estimates

.... for some of the logic described here

assuming 10 μm^2 per inverter, 20 μm^2 per 2 I/P NAND/NOR (in 130nm)*

CWD logic: $\sim 16 \times 16 \mu\text{m}^2$

priority logic to select mux: $\sim 16 \times 16 \mu\text{m}^2$

mux logic for 3 per pixel: $\sim 50 \times 50 \mu\text{m}^2$

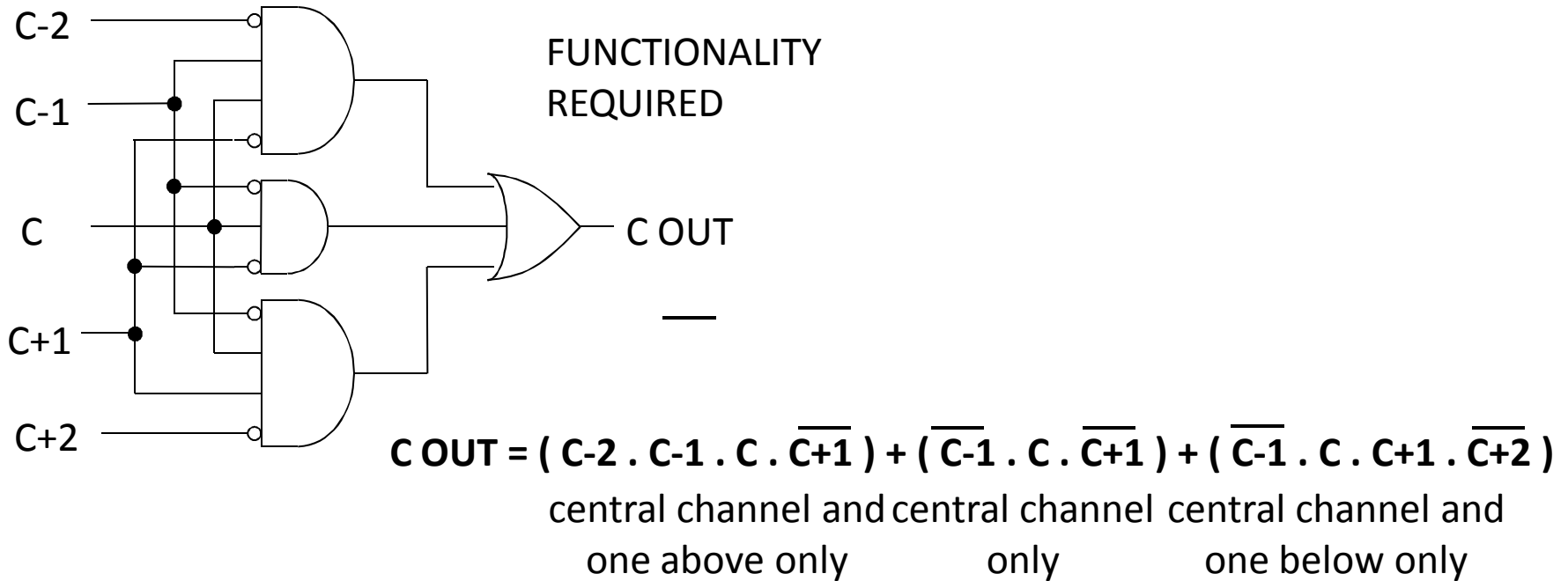
so no big area consumption here – c.f. pixel size $\sim 100 \times 1500 \mu\text{m}^2$

but much other functionality not yet included

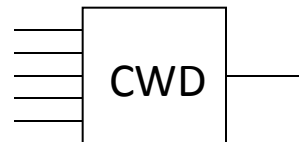
* *W. Erdmann:*

<http://indico.cern.ch/getFile.py/access?contribId=7&resId=1&materialId=0&confId=55224>

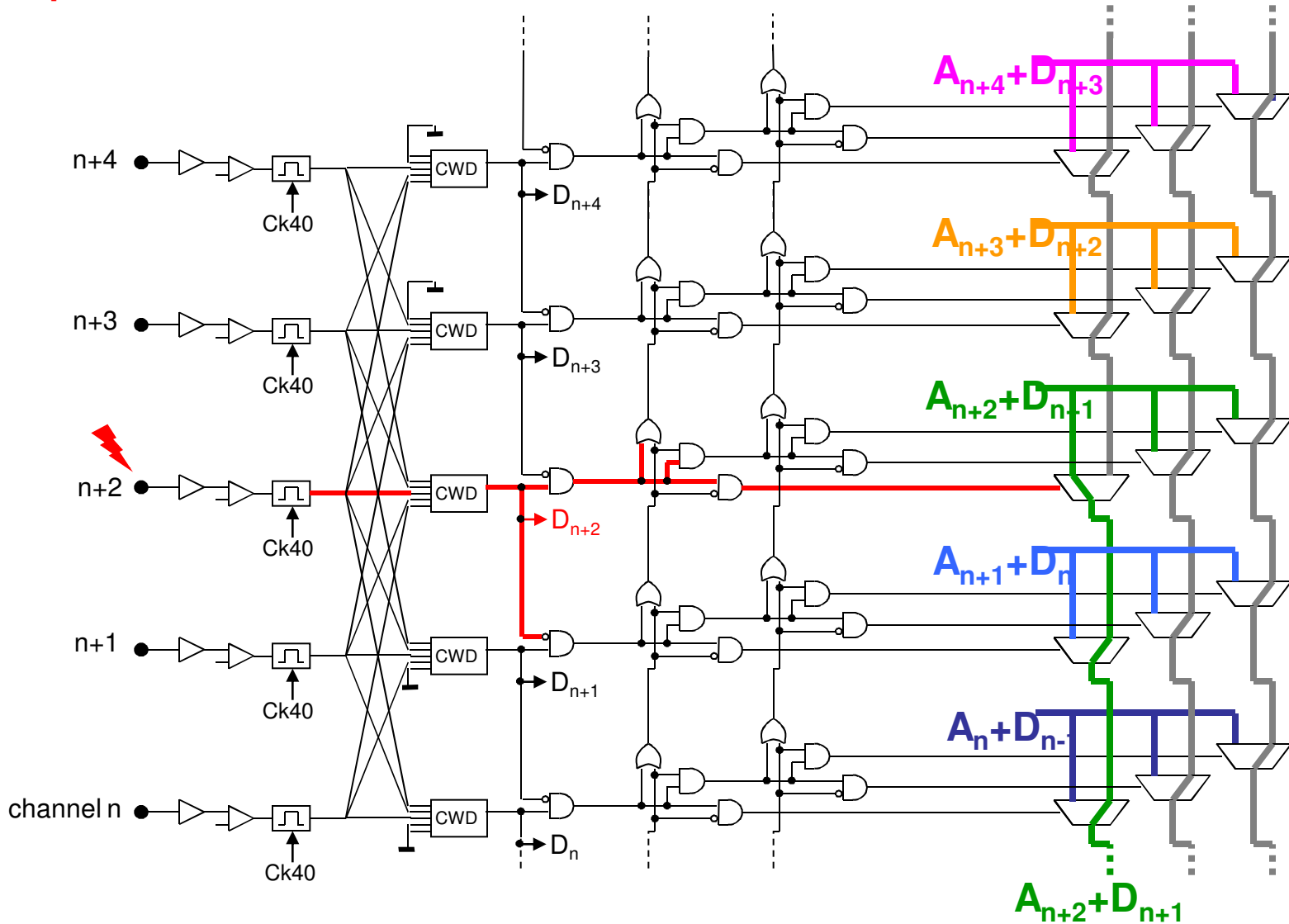
cluster width discrimination



In the diagrams the cluster width discrimination logic above is represented by

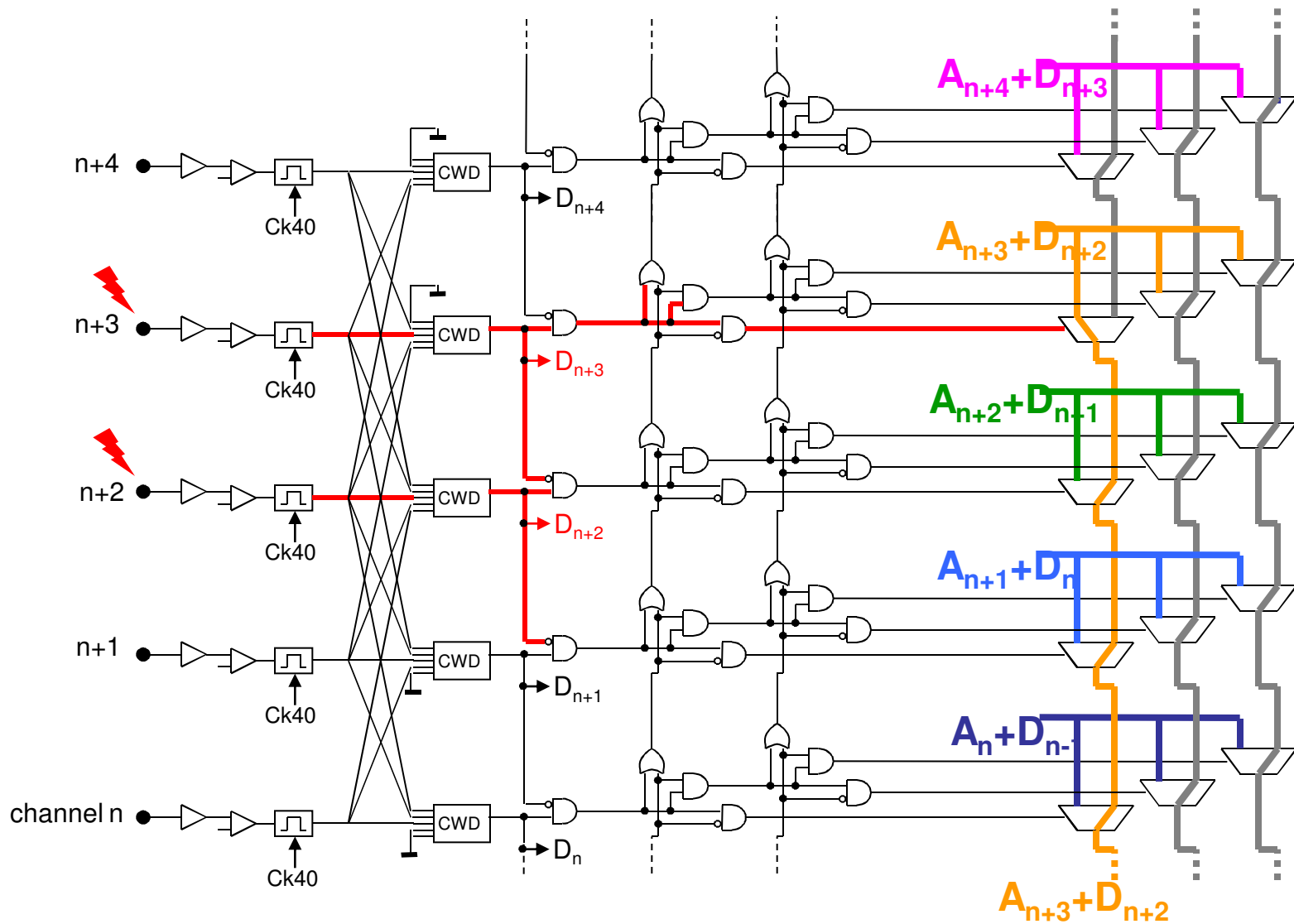


example - one hit on one channel



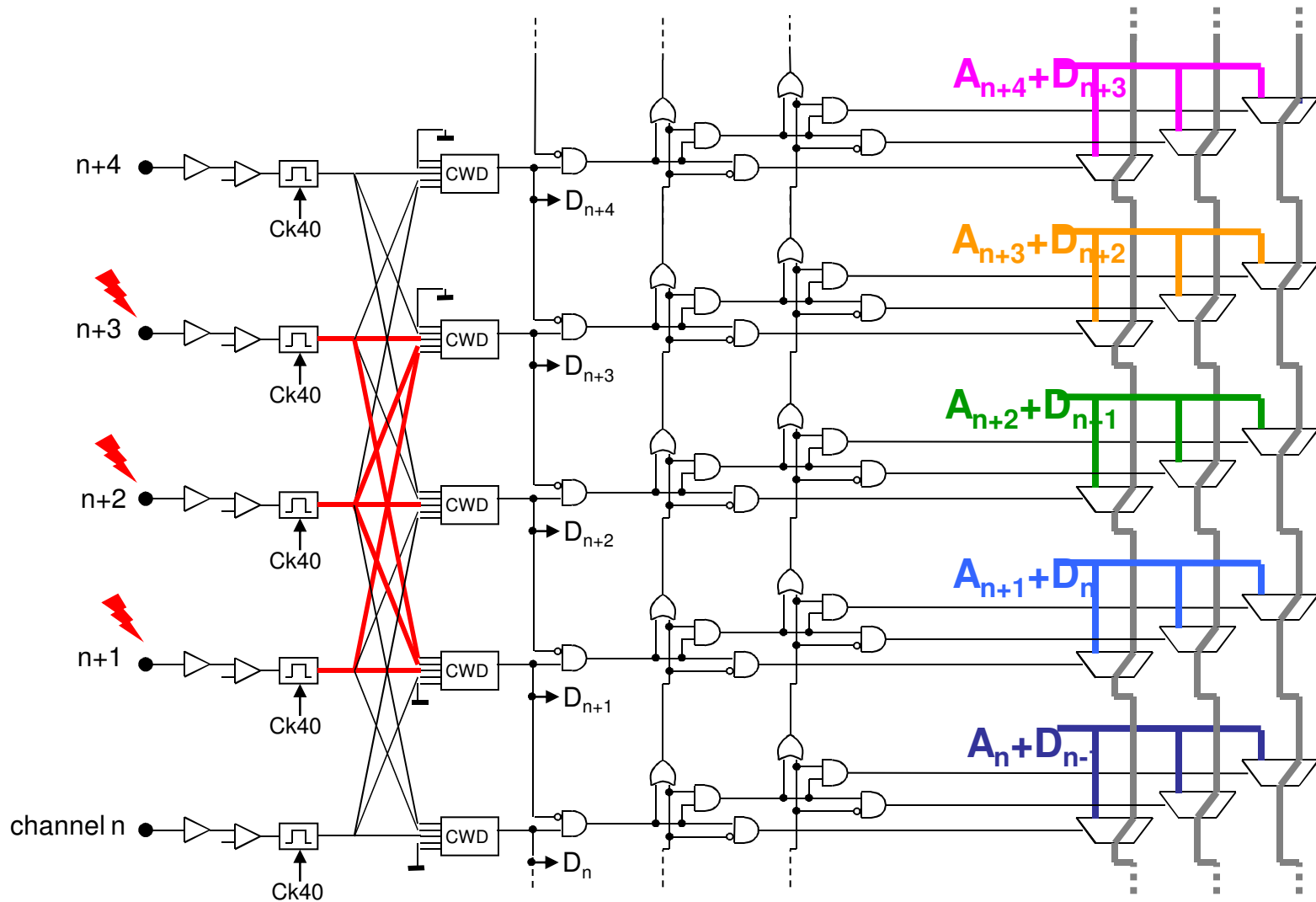
25 nsec pulse propagates through logic, selecting multiplexer in first column
address of hit pixel passes through multiplexer chain to end of column

one hit shared between 2 channels



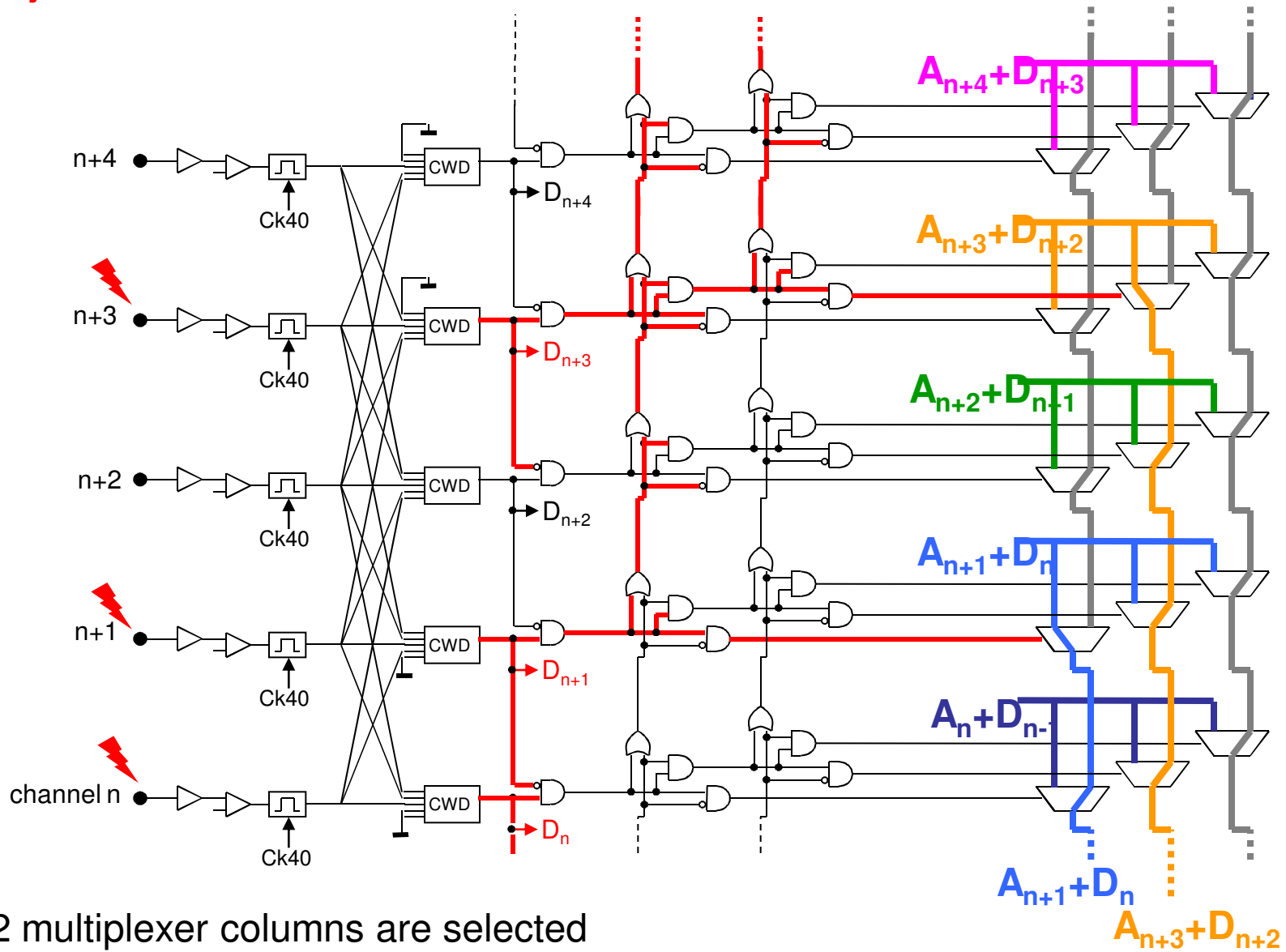
simple logic stops the lower channel signal feeding through to the mux select input
so 7-bit address of only one channel (A_{n+3}) is selected
double channel cluster indicated by including hit signal D_{n+2} from lower channel 35
(so 8 bits altogether)

one hit shared between 3 channels



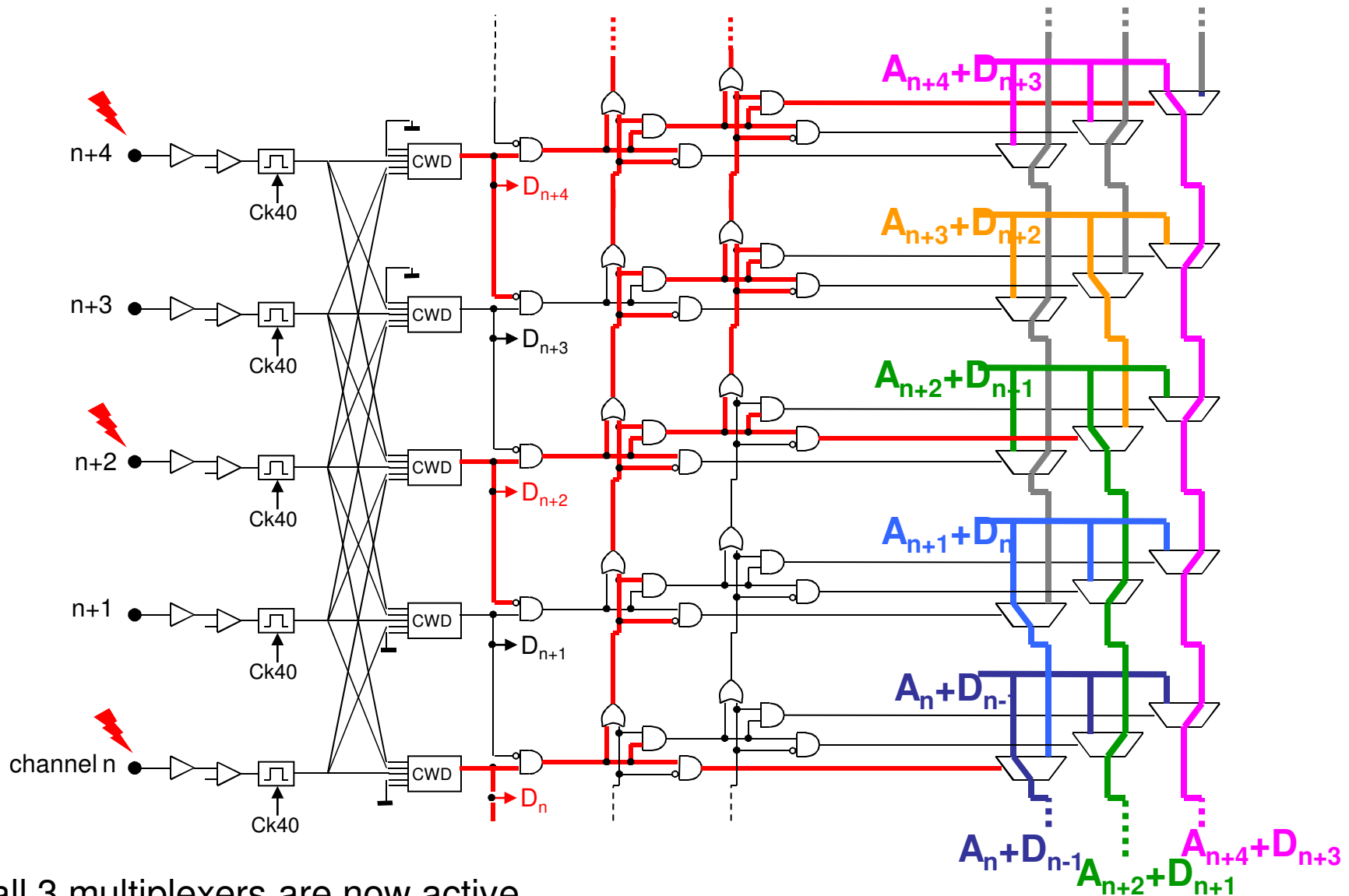
CWD logic rejects signals in 3 (or more) adjacent channels
no multiplexer is selected

2 adjacent + 1 isolated



2 multiplexer columns are selected

3 isolated hits



all 3 multiplexers are now active

a cluster further up the chip would be lost in this design

(design would have to expand if more than 3 clusters required – but linear expansion of