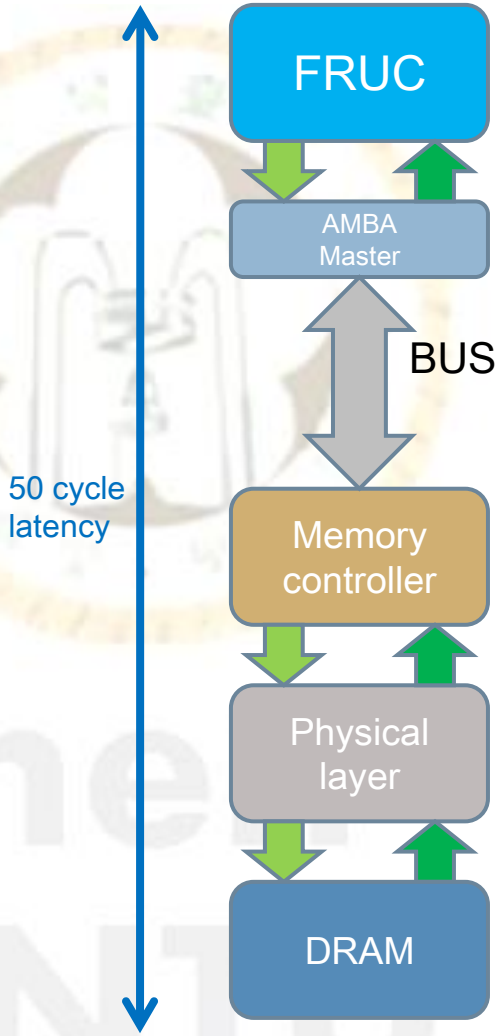
Hardware Implementation
Specification Analysis & Challenge

- Many DSPs share DRAM with system bus on LCD
  - Except FRUC
- The path between FRUC & DRAM
  - Assume 50 cycle latency (with uncertainty)
- Clock frequency
  - 300 M Hz
- I/O
  - 16 byte / cycle (128 bit BUS)
- Pixels' arrangement on DRAM
  - 4 successive pixels / address, 2 banks

0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0	13,0	14,0	15,0
0,1	1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	10,1	11,1	12,1	13,1	14,1	15,1
0,2	1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	10,2	11,2	12,2	13,2	14,2	15,2
0,3	1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	10,3	11,3	12,3	13,3	14,3	15,3



Hardware Implementation
Specification Analysis & Challenge

- DRAM selection
  - Reference to NXP 5100 FRUC solution
    - Output frame: 1920x1080, 120Hz
    - 2pcs 16bit x 512 Mb DDR2-667
  - Our target
    - Output frame : 3840x2160, 120Hz (four times)
    - 4pcs 16bit x 1Gb DDR3-1333 (each bank = 2pcs)
      - Data rate & storage are four times than NXP 5100
    - The price gap between DDR2 & DDR3 becomes closer

DRAM Spot Price MORE — LastUpdate:Nov.3 2010, 18:00 (GMT+8)

Mode	Item	Daily High	Daily Low	Session High	Session Low	Session Average	Session Change	History
	DDR3 2Gb 256Mx8 1333MHz	3.10	2.88	3.05	2.88	2.96	(-0.87%)	
	DDR3 1Gb 128Mx8 1333MHz	1.80	1.63	1.78	1.63	1.69	(-0.71%)	
	DDR3 1Gb 128Mx8 eTT	1.62	1.50	1.62	1.50	1.55	– (0.00%)	
	DDR3 1Gb 128Mx8 eTT (Quasi)	1.45	1.30	1.45	1.30	1.35	– (0.00%)	
	DDR2 1Gb 128Mx8 800MHz	1.72	1.55	1.67	1.55	1.59	(-4.09%)	
	DDR2 1Gb 128Mx8 eTT	1.67	1.57	1.62	1.57	1.60	(-3.51%)	
	DDR 512Mb 64Mx8 400MHz	1.67	1.58	1.65	1.58	1.61	(-1.59%)	

DRAM Contract Price (Oct. 2H) MORE — LastUpdate:Oct.29 2010, 16:37 (GMT+8)

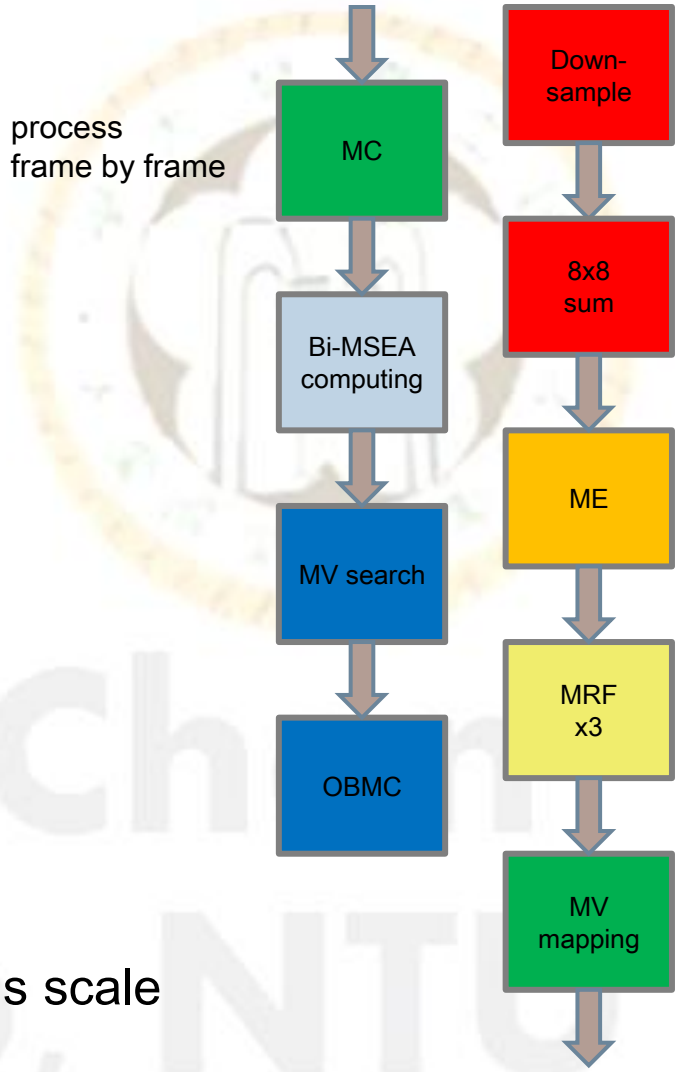
* Reference from http://www.dramexchange.com/

Hardware Implementation
Specification Analysis & Challenge

- Resource available
  - Max. bandwidth
    - 1333MHz *4pcs*16bits / 8bits = 10666 MB/s
    - DRAM random access penalty
      - Assume **65%** probability of request failure
    - 10666 x (100% - 65%) = 3733.3 MB/s
    - 3733.3 / 24 fps = **155.6 MB / frame – 24Hz**
    - 3733.3 / 60 fps = **62.2 MB / frame – 60Hz**
  - Max. cycles
    - 300 MHz / 24 fps = **12.5 M cycles / frame – 24Hz**
    - 300 MHz / 60 fps = **5.0 M cycles / frame – 60Hz**

Hardware Implementation
Specification Analysis & Challenge

- Challenges
  - Cycles
    - Pipeline bubble reduction
    - Data pre-fetch
      - Dependency problem
  - Area
    - Hardware re-use
      - Many different operations
  - Bandwidth
    - Support reading & writing up to
      - 13 frames for 24 Hz
      - 5.2 frames for 60 Hz
  - SRAM
    - Search range ±128 x ±128



- For limited bandwidth & cycles
  - All pixel comparisons operate at 1080p's scale
    - MSEA, MRF energy, bi-MSEA, BE.

Hardware Implementation - ME
Ping-pong & two-way scheduling

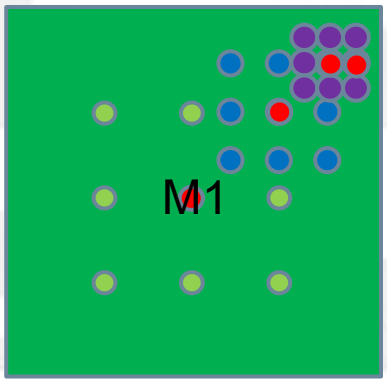
* 2040 : total # of 32x32 blocks in a frame

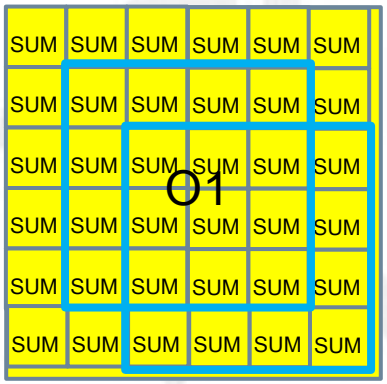
- SRAM issue
  - Scheme C (save whole search range), ±128 x ±128
    - SRAM size = (128+128+32)^2 = **82944 Byte** ~= **1.3 M gate count**
    - Bandwidth = 2MB x (128+128+32) / 32 = **18MB / ME**
    - Cycle = (128+128+32) x 32 / 16 = **576 cycles / block**
  - To employ the characteristics of ME algorithm
    - 4, 2, 1-step convergence
      - Only call the required pixels
    - 8x8 MSEA criterion, 8-step from origin for re-estimation
      - Apply Scheme C for 8x8 sum with much less SRAM & bandwidth
    - Bandwidth = (2304x60% + 2304x2x40% + 36x4x2)x2040 = **7.2 MB / ME**

Save all pixels in convergence's range

Search range
±8 x ±8

Required size
(8+8+32)^2
= **2304 Byte**





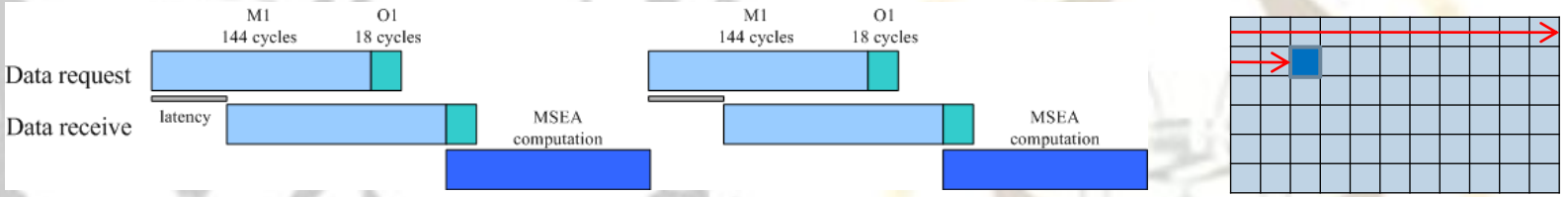
Save all 8x8 sum in search range

Search range
±128/8 x ±128/8
= **±16 x ± 16**

Required size
(16 + 16 + 4)^2 x 2 Byte
= **2592 Byte**

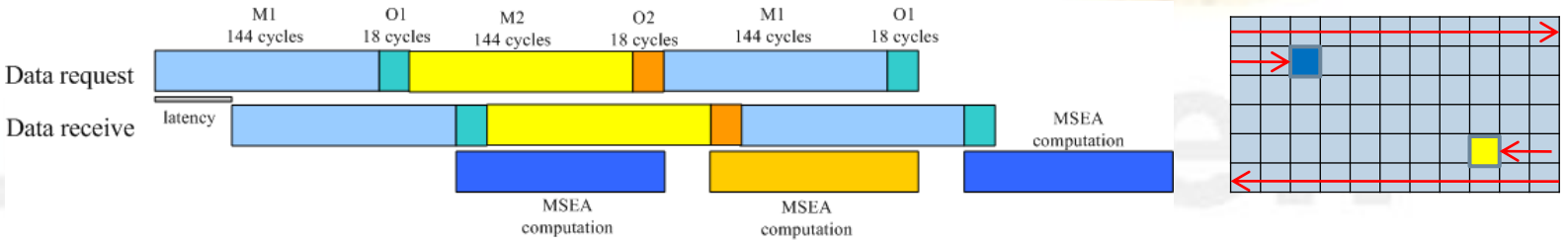
Hardware Implementation - ME
Ping-pong & two-way scheduling

- Scheduling issue
  - Directly scheduling



    - Lot of bubbles

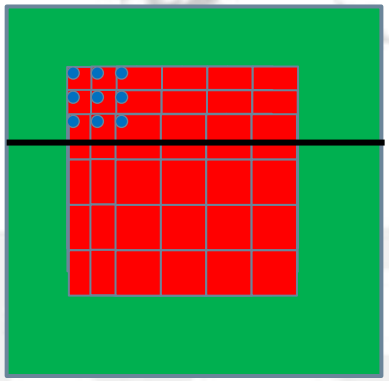
  - With additional SRAM pair (M2 & O2) for ping-pong usage & two-way scheduling

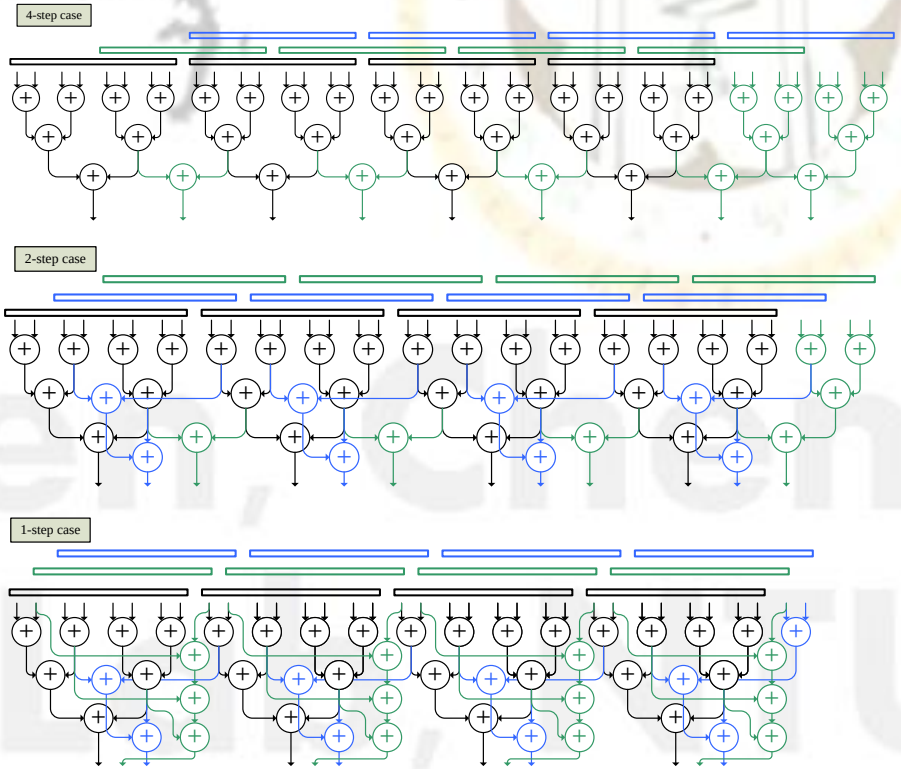


    - One is computing while the other one is fetching data
    - Cycle consumed for 4, 2, 1-step convergence = 18 + 144 = 162 for the best balance

Hardware Implementation - ME
Flexible adders

- Different adder arrangement for 4,2,1-step converge
  - 8x8 sum generating
    - Read 1 line of pixels in M1 (or M2)
    - Generate 8x1 sums from sum-trees for all 9 candidate per cycle



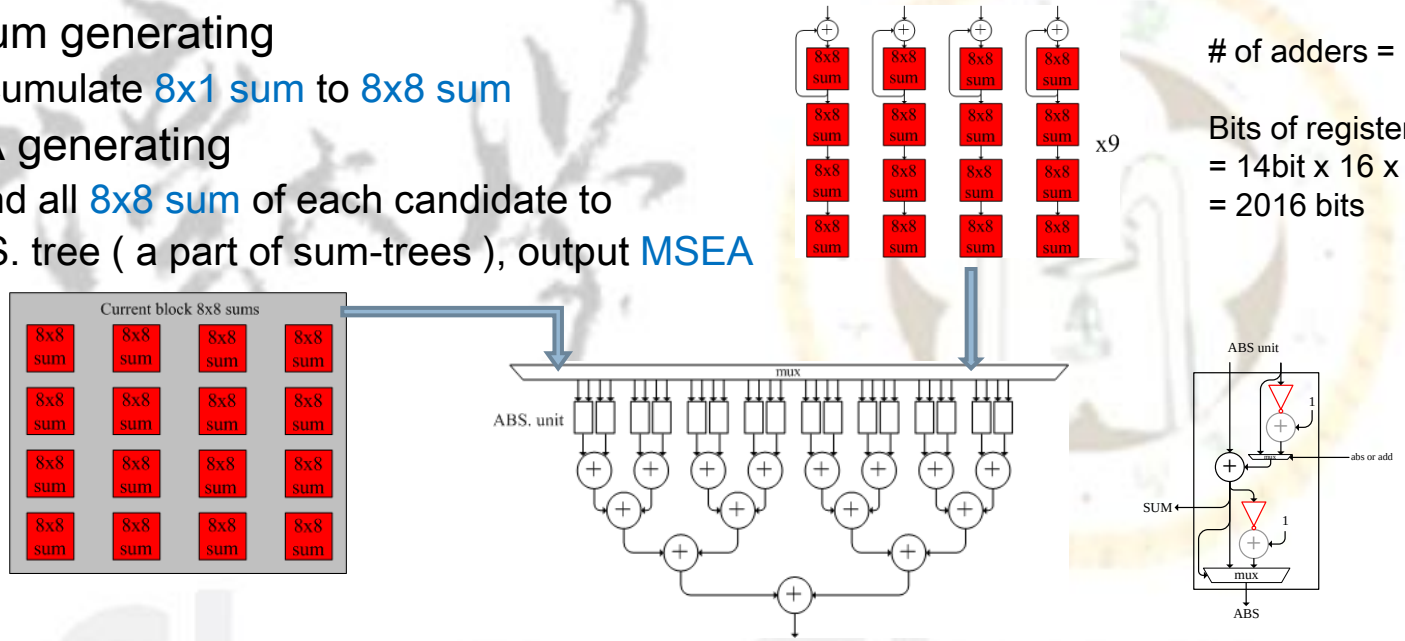


Sum-trees

# of adders = 49

Hardware Implementation - ME
Flexible adders

- Different adder arrangement for 4,2,1-step converge
  - 8x8 sum generating
    - Accumulate 8x1 sum to 8x8 sum
  - MSEA generating
    - Send all 8x8 sum of each candidate to ABS. tree (a part of sum-trees), output MSEA



Accumulators

# of adders = 36

Bits of registers
= 14bit x 16 x 9
= 2016 bits

- For re-estimate, directly send existing 8x8 sum from O1 (or O2) to abs. tree
- # of cycles
  - For 4,2,1-step : (40 + 9 + 4) + (36 + 9 + 4) + (34 + 9 + 4) = **149 cycles**
  - For 8-step : (5 + 4) or (3 + 4) = **9 or 7 cycles**
  - The worst = 149x60% + (149+128/8x9 + 149)x40% = **266 cycles / block**

Hardware Implementation - ME
Cost Analysis

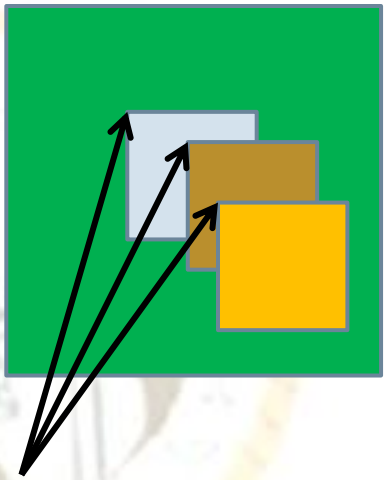
- Cycles
  - Schedule is tight
  - 576 cycles / block → **266 cycles / block**
  - # of cycles consumed is also balanced with data fetch
- Area
  - 341 adders → 49 + 36 = **85 adders**
  - Sum-trees is also used for down-sample, 8x8 sum on whole frame, MRF energy, bi-MSEA & boundary error computing
- Bandwidth
  - 18MB / ME → **7.2 MB / ME**
- SRAM
  - M1 & M2 : 48 address x 128 bits x 3Banks x 2 = 4608 Byte
  - O1 & O2 : 84 address x 16 bits x 16Bank x 2 = 5376 Byte
  - 82944 Byte → **9984 Byte**
  - Also shared by all the following modules

Throughput
149 cycles / 25 candidates
~= 6 cycles / candidate

SAD + 2D adder trees for the same throughput
(32x32x2) / 6 ~= **341 adders**
More SRAM banks

Hardware Implementation - MRF correction
MV grouping for data reuse

- Bandwidth issue
  - Fetch all the 8 neighboring candidates' pixels for MSEA computing
    - Bandwidth ~= 2MB x 8 = **16MB / iteration**
    - Cycle = 64 x 8 = **512 cycles / block**
  - To employ the characteristics of MVF
    - Neighboring MVs are similar, many pixels required are overlapped
    - To group them for data reuse
      - Group size >= 3 for bandwidth gain
      - At most 2 groups

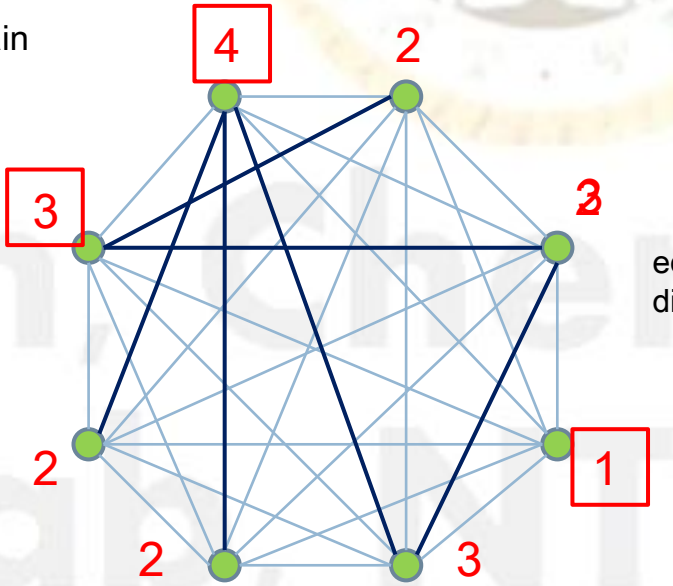


M1 or M2

Search range
±8 x ±8

MRF energy =

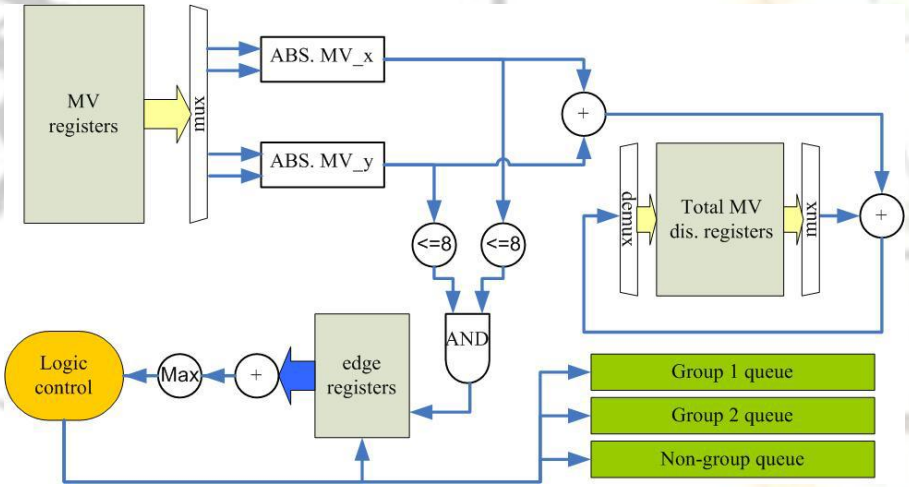
$$\mathrm{MSEA}_{candidate} + weight * \sum_{neighbor} |\mathrm{MV}_{candidate} - \mathrm{MV}_{neighbot}|$$



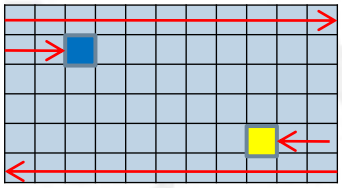
edges that discontinuity <= 8

Hardware Implementation - MRF correction
MV grouping for data reuse

- Grouping architecture



  - Also used for ME's median filter
  - MSEA computing
    - By ME's sum-trees & accumulators
    - To fetch & data compute according to queue
    - Write out 9 candidates' MSEA to DRAM for re-use
- Scheduling
  - Ping-pong & two-way, like ME



Hardware Implementation - MRF correction
MV grouping for data reuse

- Group results of MRF iteration 1

Average # of blocks (total 2040)

Average # of candidates (total 2040 x 8 = 16320)

	Group1 size8	Group1 size7	Group1 size6	Group1 size5	Group1 size4	Group1 size3	Group2 size4	Group2 size3	non-group
park_joy	1291	113	109	242	135	112	17	160	1912
ducks_take_off	1771	130	66	35	21	12	1	8	521
vintagecar	998	237	209	274	191	112	20	202	2267
tractor	1314	196	140	198	135	53	32	172	1269
pedestrian_area	1328	174	147	151	127	74	13	98	1761
transformer 7-3	1283	132	98	194	108	116	6	146	2338
Titanic-2	945	312	213	286	152	95	13	222	2268

  - Choose transformer 7-3 as the worst case
    - Cycle / block ~= (32 + 4) x (16320 - 2338) + 64 x (2338) / 2040 = **320 cycles / block**
    - Bandwidth ~= 48 x 48 x (G1# + G2#) + 32 x 32 x (NG#)
      = 4.8 MB + 2.4 MB = **7.2 MB**
- For MRF iteration 2 & 3
  - The MVF changes a little
  - Load previous iteration's 9 MSEA results, compute the necessary MSEA
    - MRF iteration 2
      - Cycle / block = **82**, bandwidth = **1.1 MB**
    - MRF iteration 3
      - Cycle / block = **57**, bandwidth = **0.3 MB**

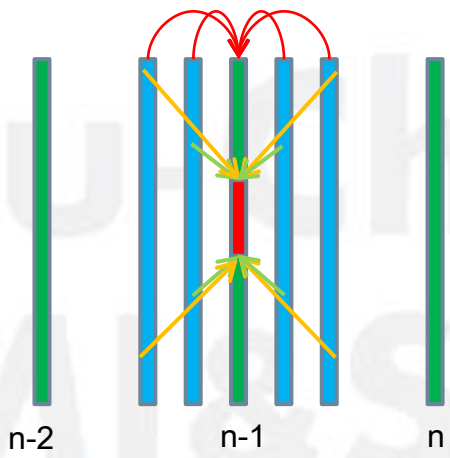
Computed by the simulation of the worst case : Titanic-2

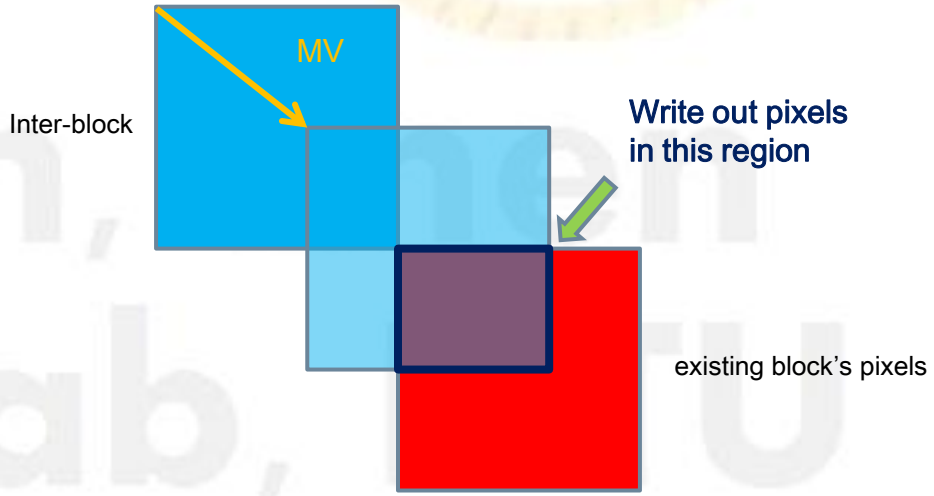
Hardware Implementation - MRF correction
Cost Analysis

- Cycles
  - Schedule is tight
  - MV grouping for data fetching reduction
  - Employ the computed results
  - 512 x 3 = 1536 cycles / block for 3 iterations → 320+ 82 + 57 = **459 cycle / block**
- Area
  - Share computation unit with ME
- Bandwidth
  - MV grouping for bandwidth reduction
  - Employ the computed results
  - 16MB x 3 iterations = 48 MB → 7.2 + 1.1 + 0.3 = **8.6 MB**
- SRAM
  - Shared with ME

Hardware Implementation - MC
Inverse-MC scheduling

- Bandwidth issue
  - Tradition MC
    - Process block by block on inter-frame, read required pixels then interpolate
    - 24Hz to 120Hz bandwidth = (3840x2160x1.5)x(4+4) = **99.5MB**
    - 24Hz to 120Hz cycles = (3840x2160x1.5)x(4+4)/16 = **6.2M cycles**
  - Inverse-MC scheduling
    - Process block by block **on existing frame**
      - Read one block's pixels of existing frame
      - For all the possible inter-blocks
        - Derive the overlapped region along their MV
        - write out pixels belong to them.
    - Bandwidth will be close to min. requirement



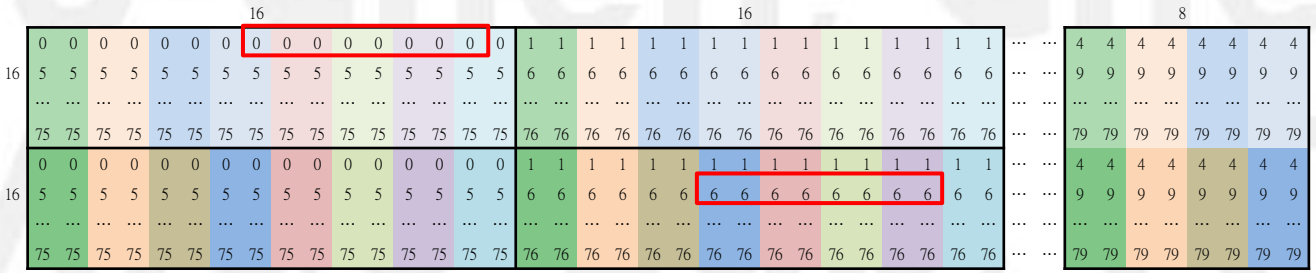


Hardware Implementation - MC Inverse-MC scheduling

- SRAM issue
  - Need to random access 8x2 pixels of existing block at a time
  - Existing block's size = (64+8)x64 = 72x64
    - For writing pixels across existing block's boundary
  - Pixel arrangement
    - Odd line : M1 or M2
    - Even line : O1 or O2
    - Random access 8x1 pixels in M1, O1, M2, O2 at a time

M1 or M2

O1 or O2

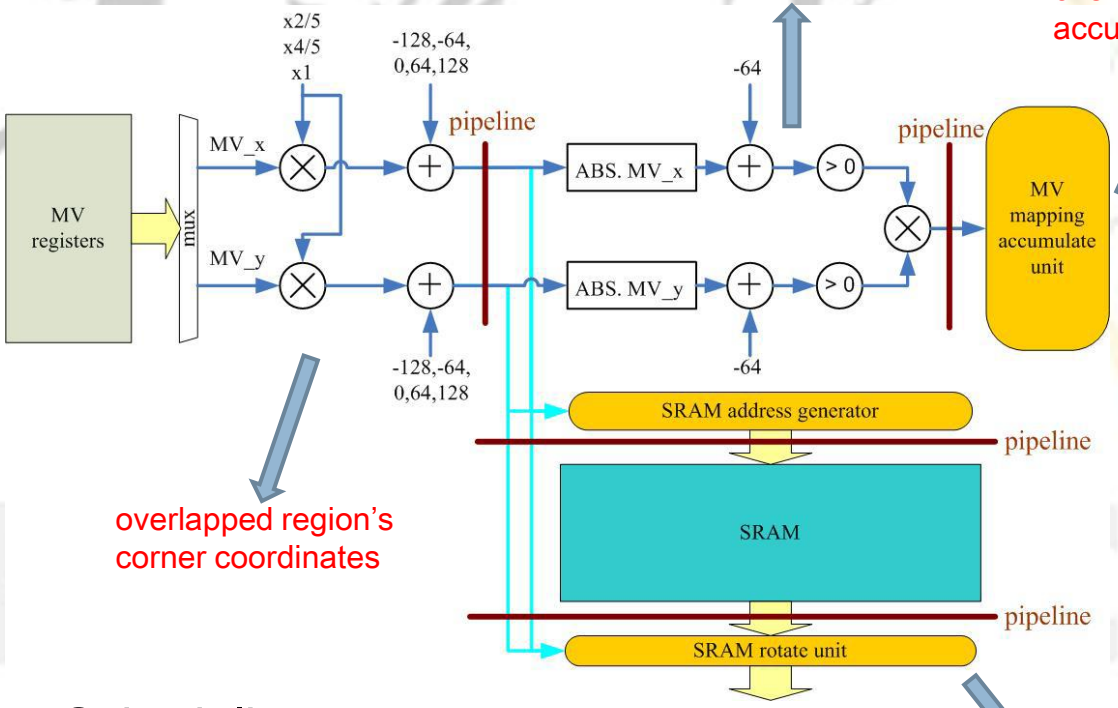


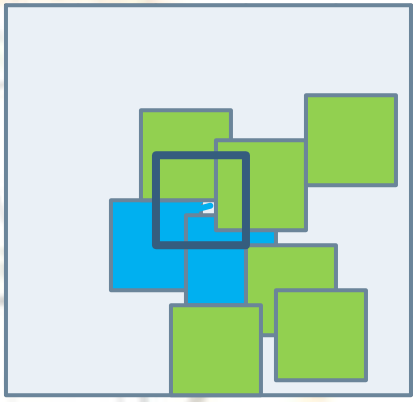
Color
different bank

Number
address

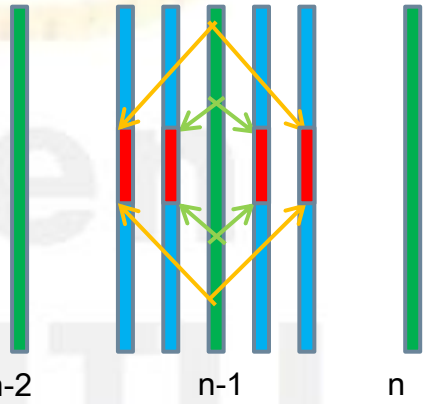
Hardware Implementation - MC
Inverse-MC scheduling

- Inverse-MC architecture
  - Also used for MV mapping





Inter frame



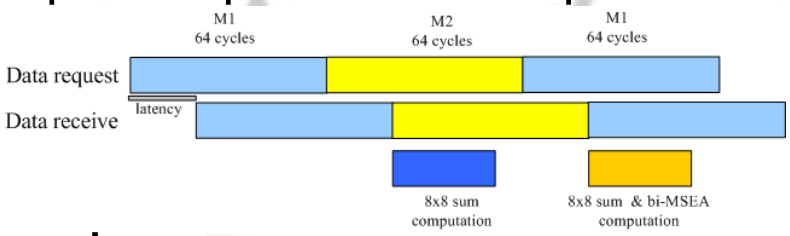
- Scheduling
  - Ping-pong SRAM usage
  - Write all Y first, the write U + V

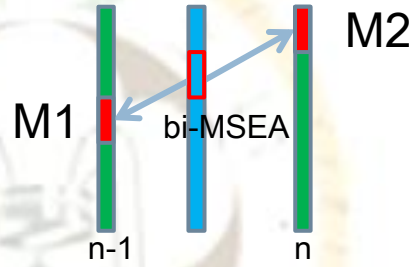
Hardware Implementation - MC
Cost Analysis

- Cycles
  - Schedule is tight
  - 6.2 M cycles → (72x64+40x32x2+64x64x1.5x4)x2040/16 = **4.0 M cycles**
  - Close to min. requirement
- Area
  - Computation unit is shared with MV mapping
- Bandwidth
  - 99.5MB → (72x64+40x32x2+64x64x1.5x4)x2040 = **64.8 MB**
  - Close to min requirement
- SRAM
  - SRAM is shared
  - Pixel arrangement

Hardware Implementation - Post-processing
Bi-MSEA, MV search, OBMC

- DRAM queue pushing & popping
- Bi-MSEA
  - Shares sum-trees & accumulators with ME
  - Ping-pong SRAM usage
  - Pre-pop next queue for data pre-fetch





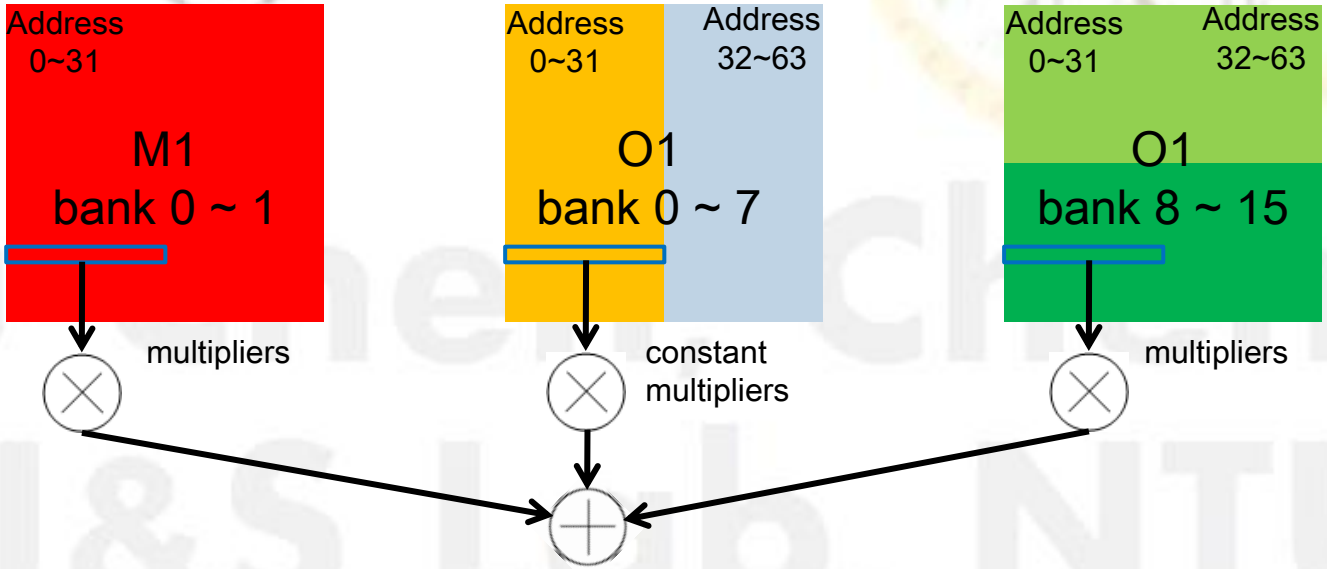
- MV search
  - Parallelism analysis
    - Search range
      - even point candidates in ±8 x ±8
  - Shares sum-trees with ME
  - SRAM interleave

Cycles left after all of the previous operations

(all of the worst cases)	cycle left (60Hz)	cycle left (24Hz)	# of sub-block (24 Hz)	cycle left for one sub-block (24 Hz)
park_joy	1126796	4462482	2474	1804
ducks_take_off	1219792	4718998	1340	3522
vintagecar	1049260	4258462	3074	1385
tractor	1191664	4641346	1969	2357
pedestrian_area	1115964	4410934	2787	1583
transformer 7-3	1048000	4199558	3947	1064
Titanic-2	1039672	4246090	1820	2333

Hardware Implementation - Post-processing
Bi-MSEA, MV search, OBMC

- OBMC
  - Save center pixels in M1 (or M2), the others are in O1 (or O2)
  - Access 16 pixel-line at a time of each parts
  - Follow by (constant) multipliers
    - constant for pixels of left & right parts
  - Fuse multiplied pixels & write out



Hardware Implementation - Post-processing
Cost Analysis

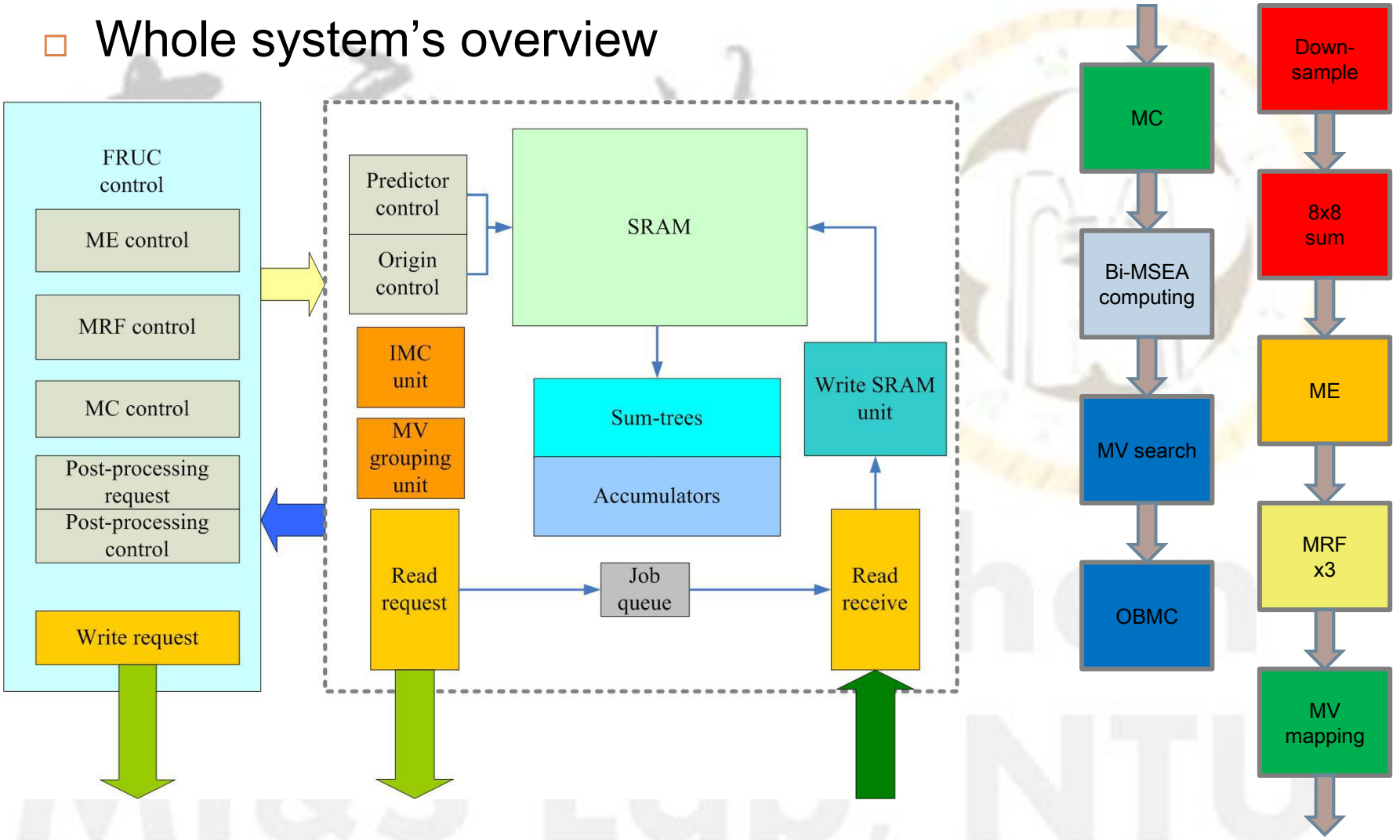
- Cycles
  - Schedule is tight
  - Can cope with
    - At least 4069 sub-blocks for 24 Hz
    - At least 1015 sub-blocks for 60 Hz
- Area
  - Parallelism analysis for min. area cost
  - Share computation unit with ME
- Bandwidth
  - Only read & write the required data
- SRAM
  - SRAM is shared
  - Pixel arrangement

Hardware Implementation
Cost Analysis Conclusion

	ME		MRFx3		MC		Post-processing	
	Direct	Proposed	Direct	proposed	Direct	proposed		
Cycles	576 cycles /block	266 cycles /block	1536 cycles /block	459 cycles /block	6.2M cycles -24Hz	4.0M cycles -24Hz	4069 sub-block -24Hz	1015 sub-blocks -60Hz
	all of the schedules are tight							
Area	sum-trees & accumulators are shared				shared with MV mapping		parallelism analysis	
Bandwidth	18MB	7.2MB	48MB	8.6MB	99.5MB	64.8MB	Only read & write the required data	
SRAM	Shared by all modules							
	Pixel arrangement							
	82944 Byte	9984 Byte						

Hardware Implementation
Implementation results

- Whole system's overview



Hardware Implementation
Implementation results

- Specification

Design Specification	
Technology	UMC 90nm
Clock rate	300MHz
Bus width	128 bits/cycle
DRAM	DDR3-1333
Gate count with SRAM	537652
Gate count without SRAM	273845
SRAM size	9984 Bytes single port

Capability Specification	
FRUC mode	24 Hz -> 120 Hz 60 Hz -> 120 Hz
Frame size	3840x2160
Search range	±128 x ±128

- Algorithm vs. hardware
  - Raster scan ⟶ two-way scan
  - MV search ±8 x ±8 ⟶ ±8 x ±8 on even points