CS 3725 March 1, 2012

Q// How to Map a matrix onto multiple memory modules such that we can access various vectors in optimal time.

Mapping function is called skewing scheme.

eg// Consider a simple mapping, $\emptyset$ which maps a 4x4 matrix A onto 4 memory modules (M_0, M_1, M_2, M_3):

$\emptyset : A[i,j] \rightarrow \{M_0, M_1, M_2, M_3\}$

such that $\emptyset(i,j) = j \quad 0 \le j \le 3 \quad 0 \le i \le 3$

row # col # index of memory module

M-Matrix (mapping onto the memory module)

M_0	M_1	M_2	M_3
A_{00}	A_{01}	A_{02}	A_{03}
A_{10}	A_{11}	A_{12}	A_{13}
A_{20}	A_{21}	A_{22}	A_{23}
A_{30}	A_{31}	A_{32}	A_{33}

↔

Q-Matrix

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

Diagonal mapping

Row-wise mapping

Module #

Row Access Any row can be accessed in 1 cycle

Diagonal Access 1 Cycle

Column Access Due to conflict, it takes at most 4 cycles.

Eg.// Consider

$$\emptyset(i,j) = (i+j+1) \bmod 4$$

(We are mapping a 4x4 matrix onto 4 memory modules)

In general, NxN matrix mapped onto N modules.

1	2	3	ø
2	3	ø	1
3	ø	1	2
ø	1	2	3

Rows: Conflict free
ie. In each can be accessed in 1 cycle

Columns: Conflict free

Diagonals: Not conflict free, takes 2 cycles.

Observations

① There does not exist any linear ø function such that an NxN matrix can be mapped onto N memory modules such that all rows,

columns and diagonals can be accessed conflict free.

② Non-linear functions are no more powerful than linear functions.

Ex: (Prime memory module)

- Uses 5 memory modules for a 4x4 matrix.

$$\emptyset(i,j) = (i+j) \bmod 5 + 1 \qquad 0 \le i,j \le 4$$

M_0	M_1	M_2	M_3	M_4
	A_{00}	A_{01}	A_{02}	A_{03}
A_{13}		A_{10}	A_{11}	A_{12}
A_{22}	A_{23}		A_{20}	A_{21}
A_{31}	A_{32}	A_{33}		A_{30}

$\emptyset$ - matrix

i \ j →	0	1	2	3
0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2

Rows: Can be accessed conflict free.

Columns: Conflict free

Diagonal: Conflict free

Drawbacks

① Modulo prime is very expensive whether at the compile stage or hardware stage.

② It has "holes", and hence less than 100% memory utilization.

③ In most machines the number of modules is a power of 2.

Multiskewing Scheme

$\emptyset_1$:
$\emptyset_2$:

N×N matrix onto N modules.

$\emptyset_1$: $(i,j) = (j - 2i) \bmod N+1$ if $i \leq N/2$
$\emptyset_2$: $(i,j) = (2i - j - \frac{N}{2} - 1) \bmod N+1$, $i > N/2$

Eg. Mapping an 8×8 matrix to 8 memory modules

$\emptyset$ matrix

		→						
	8	(1)	2	3	4	5	6	7
$\emptyset_1$	6	7	8	(1)→	2	3	4	5
	4	5	6	7	8	(1)→	2	3
	2	3	4	5	6	7	8	(1)→
	5	4	3	2	←(1)	8	7	6
$\emptyset_2$	7	6	5	4	3	2	←(1)	8
	1	8	7	6	5	4	3	2
	3	2	←(1)	8	7	6	5	4

Rows: Conflict free

Columns: Conflict free

Diagonal : Conflict free

- No prime arithmetic
- No of memory modules power of 2
- No holes