The University of Texas at Arlington

EE – 6313
Advanced Microprocessor Systems Project
Cache Controller Design

Submitted by: Rao Waqas Ali
Submitted to: Dr. Jason Losh

Introduction

The goal of this project is to determine the best architecture for a cache controller that interfaces to a 32- bit microprocessor with a 32-bit data bus. While the microprocessor is general purpose in design and performs a number of functions, it is desired to speed up certain signal processing functions, such as a fast Fourier transform (FFT) routine. 2 Detailed Requirements You are given the task of determining the architecture of cache memory to speed up a microprocessor system. After the program being executed was profiled, the largest hit in performance was seen in a function called Radix2FFT.

The project determines the best architecture for the 256 KiB cache including the associative set size (N), number of cache lines (L), and burst length (BL), and write strategy (write-back, write-through allocate, or write-through non-allocate). Your project should seek to allow 4GiB of SDRAM to be interfaced using a 32-bit data bus (arranged as 2 30 x 32-bits) and the cache should be limited to 256 KiB in size. The size of the FFT will be 32768 points (512KiB) in normal operation.

3 configurations result based on associativity and burst length and the write strategy are presented in this report.

The speed is measured on the basis of keep count of the counters and then using the formula to calculate the Write Hit Count and Read Hit Count and also the sum of the two.

Specifications

Maximum number of lines (L) = 655356.
Cache Size = 262144.
Maximum number of ways (N) = 16.

Data Width = 32 bits.
Address = 32 bits.

Cache Size (S) = (N_Lines (L) * N_Ways (N) * Burst Length (BL) * Data Width / 8))

Block Size = Burst Length * (Data Width / 8).

The formula can be rearragned to find other values.

log2(L) = line bits.
log2(Block_Size) = b bits

If N = 1, it’s a direct mapped cache. Else N associate.
If L = 1, it’s a Fully Associate cache.

Logic Blocks

1. Calculations (Line, Tag)

The line and tag are calculated using the ***uint32_t LINE(uint32_t)*** ***uint32_t TAG(uint32_t)*** functions respectively. In calculation of the line, the given address is shifted left by tag bits and then shifted right by the (tag bits + block bits(b)).

In the calculation of tag, the address is shifted right by (line bits + block bits (b)).

2. Updation of Least Recently Used Block

The current LRU value is calculated first (just N - 1) and then we iterate through the associativity and see if the LRU is less than the current. If yes, we age the LRU index and the current LRU will become zero.

```
void LRU_update(uint32_t indexx, uint32_t line)
{
    // N is 1 does not make any sense
    int i;
    uint32_t current_lru_value =  LRU_MATRIX[indexx][line]; // store the current value
    // zeroing th current value and adding 1 (age) to the other values
    // current_lru_value = 0;
    LRU_MATRIX[indexx][line] = 0;
    for (i = 0; i < N_Ways; i++)
    {
        if (LRU_MATRIX[i][line] < current_lru_value)
        {
            LRU_MATRIX[i][line] = LRU_MATRIX[i][line] + 1;
        }
    }

}
```

4. Hit Or Miss

If the tag matches the given tag, it’s a hit else it’s a miss.

```
bool hit_or_miss(uint32_t tag, uint32_t line)
{
    hit_aur_miss = false;
    uint32_t help;

    for (int i = 0; i < N_Ways; i++)
    {
        help = TAG_MATRIX[i][line];
        if(help == tag)
        {
            hit_aur_miss = true;
            break;
        }
    }
    return hit_aur_miss;
}
```

3. Read Memory

The old line is initialized to the garbage value. The tag and line is calculated and we dive into the read block section to read the block. Then we make the current line equal to the old line.

4. Read Block

The read block count is incremented, then the LRU index is calculated. The condition is checked if it was a miss or hit, in case of hit the Read Hit Counter is incremented and the LRU is updated. In case of a miss, the block state is checked along with the dirty bit to see if the other counters should be incremented or not. The read miss counter here is incremented always irrespective of the dirty and valid bits in case of a miss. The tag is updated and after that the LRU is updated as always.

5. Write Memory

There is technically no difference in the function write memory and read memory. The algorithm us the same, however the write block differs with the read block.

6. Write Block

The write block count is incremented. The write LRU index is calculated using the calculate LRU function. The hit or miss is evaluated. If it is a miss and the strategy is write back or write through allocate and the block is not free and the dirty bit is set the write block replace count and dirty replace count counters are incremented. LRU and tag are updated. Valid bit becomes one. In case of a hit, the write hit count is increased and LRU is updated. If it's a writeback, the dirty bit becomes one, at the end if it's a write through non allocate the the write block miss count is incremented (miss case).

Write Strategy

WriteBack

When a cache controller uses a writeback policy, **i**t writes to valid cache data memory and **not to main memory**. In this strategy the data is written back to cache in case of a hit, in case of a miss, the block of data gets replaced using the LRU method.

Write Through Allocate

In this strategy, we **do not care about the hit or miss**, the data is written back to the main memory.

Write Through Non Allocate

This strategy kind of combines the two. It will write to memory, however in case of the hit it writes back to the cache **making it slow.**

Formulae

Multiply with 60 ns. In hit case multiply with 1 is not shown for obvious reason.

- Write Hit Ratio = (Write Hit Count / (Write Hit Count + Write Miss Count))
- Read Hit Ratio = (Read Hit Count / (Read Hit Count + Read Miss Count))
- Total Hit = Write Hit Ratio + Read Hit Ratio
- Total Miss = (1 - Read Hit Ratio) + (1 - Write Hit Ratio)
- Total Time WTNA = Read Time Taken + Write time for WTNA + (FC * 6)

Results

1. Test Container (Initial Test)

When the test container was run on the code, **the WB and WTA always came up to be the best ways, and WTNA falling behind.** The burst length and associativity did not make a significant difference in case of just the test container.

N_Ways	BL	STR	RMC	RBC	RBHC	RBMC	RBR	RBDR	WMC	WBC	WHT	WMC	WRC	WDC	WTC	FC	write hit ratio	rd hit ratio	TOTAL
1	1	WB	65536	65536	65536	0	0	0	131073	131073	65535	65538	2	2	0	65536	0.499988556	1	1.499988556
1	1	WTA	65536	65536	65536	0	0	0	131073	131073	65535	65538	0	0	131073	65536	0.499988556	1	1.499988556
1	1	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
1	2	WB	65536	65536	65536	0	0	0	131073	131073	98301	32772	4	4	0	32768	0.74997139	1	1.74997139
1	2	WTA	65536	65536	65536	0	0	0	131073	131073	98301	32772	0	0	131073	32768	0.74997139	1	1.74997139
1	2	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
1	4	WB	65536	65536	65536	0	0	0	131073	131073	114685	16388	4	4	0	16384	0.874970436	1	1.874970436

1	4	WTA	65536	65536	65536	0	0	0	131073	131073	114685	16388	0	0	131073	16384	0.874970436	1	1.874970436
1	4	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185
1	8	WB	65536	65536	65536	0	0	0	131073	131073	122869	8204	12	12	0	8192	0.937408925	1	1.937408925
1	8	WTA	65536	65536	65536	0	0	0	131073	131073	122869	8204	0	0	131073	8192	0.937408925	1	1.937408925
1	8	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185
2	1	WB	65536	65536	65536	0	0	0	131073	131073	65534	65539	32771	32771	0	32768	0.499980927	1	1.499980927
2	1	WTA	65536	65536	65536	0	0	0	131073	131073	65534	65539	0	0	131073	32768	0.499980927	1	1.499980927
2	1	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
2	2	WB	65536	65536	65536	0	0	0	131073	131073	98298	32775	16391	16391	0	16384	0.749948502	1	1.749948502
2	2	WTA	65536	65536	65536	0	0	0	131073	131073	98298	32775	0	0	131073	16384	0.749948502	1	1.749948502
2	2	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
2	4	WB	65536	65536	65536	0	0	0	131073	131073	114678	16395	8203	8203	0	8192	0.874917031	1	1.874917031
2	4	WTA	65536	65536	65536	0	0	0	131073	131073	114678	16395	0	0	131073	8192	0.874917031	1	1.874917031
2	4	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185
2	8	WB	65536	65536	65536	0	0	0	131073	131073	122854	8219	4123	4123	0	4096	0.937294485	1	1.937294485
2	8	WTA	65536	65536	65536	0	0	0	131073	131073	122854	8219	0	0	131073	4096	0.937294485	1	1.937294485
2	8	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185
4	1	WB	65536	65536	65536	0	0	0	131073	131073	65532	65541	49157	49157	0	16384	0.499965668	1	1.499965668
4	1	WTA	65536	65536	65536	0	0	0	131073	131073	65532	65541	0	0	131073	16384	0.499965668	1	1.499965668
4	1	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
4	2	WB	65536	65536	65536	0	0	0	131073	131073	98292	32781	24589	24589	0	8192	0.749902726	1	1.749902726
4	2	WTA	65536	65536	65536	0	0	0	131073	131073	98292	32781	0	0	131073	8192	0.749902726	1	1.749902726
4	2	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
4	4	WB	65536	65536	65536	0	0	0	131073	131073	114664	16409	12313	12313	0	4096	0.874810222	1	1.874810222
4	4	WTA	65536	65536	65536	0	0	0	131073	131073	114664	16409	0	0	131073	4096	0.874810222	1	1.874810222
4	4	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185

4	8	WB	65536	65536	65536	0	0	0	131073	131073	122824	8249	6201	6201	0	2048	0.937065605	1	1.937065605
4	8	WTA	65536	65536	65536	0	0	0	131073	131073	122824	8249	0	0	131073	2048	0.937065605	1	1.937065605
4	8	WTNA	65536	65536	65535	1	0	0	131073	131073	65538	65535	0	0	131073	0	0.500011444	0.999984741	1.499996185
8	1	WB	65536	65536	65536	0	0	0	131073	131073	65528	65545	57353	57353	0	8192	0.499935151	1	1.499935151
8	1	WTA	65536	65536	65536	0	0	0	131073	131073	65528	65545	0	0	131073	8192	0.499935151	1	1.499935151
8	1	WTNA	65536	65536	65535	1	0	0	131073	131073	65536	65537	0	0	131073	0	0.499996185	0.999984741	1.499980927
8	2	WB	65536	65536	65536	0	0	0	131073	131073	98280	32793	28697	28697	0	4096	0.749811174	1	1.749811174
8	**2**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**98280**	**32793**	**0**	**0**	**131073**	**4096**	**0.749811174**	**1**	**1.749811174**
8	**2**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65536**	**65537**	**0**	**0**	**131073**	**0**	**0.499996185**	**0.999984741**	**1.499980927**
8	**4**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**114636**	**16437**	**14389**	**14389**	**0**	**2048**	**0.874596599**	**1**	**1.874596599**
8	**4**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**114636**	**16437**	**0**	**0**	**131073**	**2048**	**0.874596599**	**1**	**1.874596599**
8	**4**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65538**	**65535**	**0**	**0**	**131073**	**0**	**0.500011444**	**0.999984741**	**1.499996185**
8	**8**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**122764**	**8309**	**7285**	**7285**	**0**	**1024**	**0.936607844**	**1**	**1.936607844**
8	**8**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**122764**	**8309**	**0**	**0**	**131073**	**1024**	**0.936607844**	**1**	**1.936607844**
8	**8**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65538**	**65535**	**0**	**0**	**131073**	**0**	**0.500011444**	**0.999984741**	**1.499996185**
16	**1**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**65520**	**65553**	**61457**	**61457**	**0**	**4096**	**0.499874116**	**1**	**1.499874116**
16	**1**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**65520**	**65553**	**0**	**0**	**131073**	**4096**	**0.499874116**	**1**	**1.499874116**
16	**1**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65536**	**65537**	**0**	**0**	**131073**	**0**	**0.499996185**	**0.999984741**	**1.499980927**
16	**2**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**98256**	**32817**	**30769**	**30769**	**0**	**2048**	**0.749624807**	**1**	**1.749624807**
16	**2**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**98256**	**32817**	**0**	**0**	**131073**	**2048**	**0.749624807**	**1**	**1.749624807**
16	**2**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65536**	**65537**	**0**	**0**	**131073**	**0**	**0.499996185**	**0.999984741**	**1.499980927**
16	**4**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**114580**	**16493**	**15469**	**15469**	**0**	**1024**	**0.874169356**	**1**	**1.874169356**
16	**4**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**114580**	**16493**	**0**	**0**	**131073**	**1024**	**0.874169356**	**1**	**1.874169356**
16	**4**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65538**	**65535**	**0**	**0**	**131073**	**0**	**0.500011444**	**0.999984741**	**1.499996185**
16	**8**	**WB**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**122644**	**8429**	**7917**	**7917**	**0**	**512**	**0.935692324**	**1**	**1.935692324**

16	**8**	**WTA**	**65536**	**65536**	**65536**	**0**	**0**	**0**	**131073**	**131073**	**122644**	**8429**	**0**	**0**	**131073**	**512**	**0.935692324**	**1**	**1.935692324**
16	**8**	**WTNA**	**65536**	**65536**	**65535**	**1**	**0**	**0**	**131073**	**131073**	**65538**	**65535**	**0**	**0**	**131073**	**0**	**0.500011444**	**0.999984741**	**1.499996185**

2. RadixFFT Test

Best Results

1. N = 2, BL = 4, WB
2. N = 2, BL = 4, WTA
3. N = 2, BL = 8, WB
4. N = 2, BL = 8, WTA

Worst Results

1. N = 16, BL = 4, WTNA
2. N = 8, BL = 4, WTNA
3. N = 4, BL = 2, WTNA

Excel file be emailed to you.