



Received: 2023/11/29
Revised: 2023/12/13
Accepted: 2024/02/27
Published: 2024/03/31

***Corresponding Author:**

Da-Been Lee

Dept. of RF & IIR Seeker R&D Lab, LIG Nex1
333 Pangyo-ro, Bundang-gu, Seongnam-si,
Gyeonggi-do, 13488, Republic of Korea
Tel: +82-31-326-9177
E-mail: dabeen.lee@lignex1.com

Abstract

본 논문에서는 최적화된 환경이 아닌 영상 탐색기와 유사한 환경에서 CPU 성능 평가를 수행하였다. 가속화 옵션을 적용하지 않았고 연산 데이터 크기는 영상 데이터 크기까지 변경하며 평가를 수행하였다. 수행 결과 공개된 벤치마크에서는 Intel CPU보다 높은 성능을 보였던 PowerPC 계열 CPU가 더 낮은 성능을 보였다. 이는 가속화 옵션의 사용 유무와 PowerPC 계열의 CPU는 캐시 사이즈보다 큰 데이터 연산 시 CCF 구조에서 보틀넥이 발생한 것이 원인이었다.

This study performed CPU performance analysis in a environment similar to image seeker rather than an optimized environment. Benchmark performed without applying the acceleration option and changing the size of the data to image data size. As a result, the PowerPC CPUs showed lower performance than Intel CPUs. This result is contrary to the public benchmark results. The reason is that the acceleration option was not used and PowerPC CPU has CCF bandwidth issues when using data larger than the cache size.

Keywords

PowerPC CPU 성능 시험(PowerPC CPU Performance Test),
인텔 CPU 성능 시험(Intel CPU Performance Test),
CPU 벤치마크(CPU Benchmark),
임베디드 시스템(Embedded System),
영상 탐색기(Image Seeker)

영상처리에 적합한 CPU 선정을 위한 PowerPC 및 Intel 계열 CPU 성능 분석

Performance Analysis on PowerPC and Intel Series CPUs to Select a Appropriate CPU for Image Processing

이다빈^{1*}, 김경일², 현효영², 김홍락²

¹LIG넥스원(주) PGM탐색기연구소 선임연구원

²LIG넥스원(주) PGM탐색기연구소 수석연구원

Da-Been Lee^{1*}, Kyoung-Il Kim², Hyo-Young Hyun², Hong-Rak Kim²

¹Research engineer, RF&IIR Seeker R&D Lab, LIG Nex1

²Chief research engineer, RF&IIR Seeker R&D Lab, LIG Nex1

1. 서론

대함 유도무기는 적 함정에게 큰 위협을 가할 수 있는 무기 중 하나로, 이동하는 함정을 목표로 추적하기 때문에 탐색기의 성능이 가장 중요하다. 초기 대함 유도무기에는 능동형 초고주파 탐색기가 사용되었지만 근래에는 ECM(electronic counter measure)에 대한 취약성을 보완하기 위해 더 정밀한 적외선 영상 탐색기도 사용되고 있다. 영상 탐색기의 성능을 높이기 위해서는 탐색기의 신호처리 시스템에서 영상 데이터를 빠르게 처리해야 한다. 이를 위해서는 적절한 성능의 CPU 프로세서를 선택하여 사용해야 한다.

일반적으로 CPU의 성능을 확인하는 방법은 CPU 제조사가 제공하는 데이터 시트에서 연산 성능을 확인하는 것이다. 보통 DMIPS(Dhrystone million instructions per second)나 실수 단위 연산 성능인 GFLOPs(giga floating operations per second)를 확인한다. 제조사가 데이터 시트를 통해 연산 성능을 제공하지 않는 경우에는 공개된 벤치마크 결과를 확인할 수도 있다.

제조사가 제공하는 정보와 공개된 벤치마크를 통해 CPU의 성능을 예측할 수 있으나 이들 자료는 각 CPU의 성능을 최대도로 사용할 수 있도록 데이터 크기, 가속화 옵션 등의 조건을 적용하여 테스트를 수행한 결과이다. 때문에 대용량의 영상 데이터를 처리하는 영상 탐색기, 그리고 임베디드(embedded) 환경에서 동작하여 가속화 옵션 사용이 불가능한 신호처리기와 같은 경우 CPU 성능은 최적의

환경에서 수행한 벤치마크 결과와 달라질 수 있다.

본 연구에서는 최적화된 환경이 아닌 가속화 옵션을 사용하지 못하는 환경에서 PowerPC 계열과 Intel 계열 CPU들의 연산 성능을 비교한다. 데이터 크기를 변경하며 자체 준비한 연산 시험 코드로 연산 성능 시험을 수행하여 데이터 크기 변화와 CPU 계열에 따른 성능 차이를 확인하고 성능차가 발생하는 원인을 분석한다.

2. 성능 시험

본 연구에서는 PowerPC 계열의 e6500 Core를 사용한 NXP사의 T2080과 Intel 계열의 E3845를 성능 시험에 사용하였다. NXP사의 T2080은 4개의 코어로 구성되고 1.8 GHz로 동작한다. 데이터 시트에 작성된 성능은 single-precision에 대해 full device에서 72 GFLOPs 이다[1]. Intel E3845는 동일하게 4개의 코어로 구성되고 1.9 GHz로 동작한다. Intel사에서 제공하는 데이터 시트에는 연산 성능에 대한 자료가 포함되어 있지 않으므로 일반 벤치마크 결과를 확인해보면 single-precision에 대해 51 GFLOPs의 성능을 보인다[4].

Fig. 1과 같이 간단한 정수 및 실수 연산을 수행하고 해당 연산에 걸린 시간을 측정하는 방식으로 연산 성능을 측정하였다. T2080의 경우 PowerPC 계열에서 공통적으로 제공하는 연산 가속화 옵션인 AltiVec[3]이 있으나 사전 정의한 조건과 같이 적용하지 않고 수

```
float *t_a = (float*)malloc(sizeof(float)*array_size);
float *t_b = (float*)malloc(sizeof(float)*array_size);
float *t_c = (float*)malloc(sizeof(float)*array_size);
```

```
for(i = 0; i < array_size; i++)
{
    t_a[i] = (int)(rand())/1000;
    t_b[i] = (int)(rand())/1000;
}
```

```
_Vx_ticks_t startT, endT;
```

```
startT = tickGet();
```

```
for(j = 0; j < 1000; j++)
{
    for(i = 0; i < array_size; i++)
    {
        t_c[i] = t_a[i] + t_b[i];
    }
}
```

Fig. 1. Performance test code

행하였다. 연산에 사용한 데이터 크기를 변경하면서 1,000회 반복 연산에 소요된 시간을 측정하였다.

3. 결과 및 분석

성능 시험 결과 Tables 1-2와 같이 PowerPC 계열 T2080의 연산 소요 시간이 Intel E3845보다 크게 나타났다. 이를 자세히 분석하기 위해 1회 연산에 필요한 CPU 클럭 수를 계산하여 비교하였다.

그 결과 사용 데이터 크기가 200,000보다 작을 경우 T2080은 Integer 더하기 연산에 7 cycle, Float 더하기 연산에 9 cycle이 소요되었고 Intel E3845의 경우 Integer 연산에 5 cycle, Float 연산에 8 cycle이 소요됨을 확인했다. 사용한 배열의 크기가 200,000보다 클 경우 T2080은 필요 cycle 수가 급격하게 증가했고 E3845는 이전과 비슷한 cycle 수가 필요했다.

이를 통해 데이터 시트 및 벤치마크의 결과와는 다르게 PowerPC 계열의 CPU 고속연산 옵션인 AltiVec을 사용하지 않을 경우 T2080은 E3845보다 느린 연산 속도를 보였고 큰 크기의 데이터를 연산할 경우 연산에 필요한 시간이 급격하게 증가하는 경향을 보였다. 동영상 해상도 표준 중 하나인 VGA(video graphics array)를 기준으로 사용한다고 가정하면 VGA의 영상 크기는 640×480 이기 때문에 영상 데이터 배열

Table 1. NXP T2080 performance test result

Data array	Integer calculation		Float calculation	
	Time (s)	Cycle	Time (s)	Cycle
10,000	0.039	7.0	0.050	9.0
100,000	0.390	7.0	0.501	9.0
200,000	2.215	19.9	2.424	21.8
500,000	7.977	28.7	8.341	30.0

Table 2. Intel E3845 performance test result

Data array	Integer calculation		Float calculation	
	Time (s)	Cycle	Time (s)	Cycle
10,000	0.025	4.8	0.042	8.0
100,000	0.303	5.8	0.425	8.1
200,000	0.865	8.2	0.923	8.8
500,000	2.293	8.7	2.46	8.9

의 크기는 307,200이다. T2080의 성능 저하가 발생한 배열 크기 200,000보다 VGA 영상 데이터의 크기가 크기 때문에 T2080에서 VGA 영상을 처리한다면 성능 저하를 예측할 수 있다.

T2080의 데이터 크기 변화에 따른 성능 저하 현상을 추가로 분석하기 위해 데이터 크기를 10,000 단위로 바꿔 Integer 연산 시험을 추가 실시하여 Table 3와 같이 170,000에서 180,000으로 배열 크기가 변경될 때 연산에 필요한 CPU 클럭 수가 급격하게 증가함을 확인하였다. 배열 크기가 170,000일 경우 연산에 사용되는 데이터 크기는 float 자료형 4 byte, 배열 수 3개를 고려했을 때 1.95 MB이기 때문에 T2080의 캐시 크기인 2 MB보다 큰 데이터를 처리할 때 성능 저하가 발생했음을 알 수 있었다.

Table 3. NXP T2080 extra performance test result

Data array	Data size	Calculation cycle
10,000	$10,000 \times 4B \times 3 = 0.11 \text{ MB}$	7.0
100,000	$100,000 \times 4B \times 3 = 1.14 \text{ MB}$	7.0
170,000	$170,000 \times 4B \times 3 = 1.95 \text{ MB}$	7.0
180,000	$180,000 \times 4B \times 3 = 2.06 \text{ MB}$	19.9
200,000	$200,000 \times 4B \times 3 = 2.29 \text{ MB}$	19.9
500,000	$500,000 \times 4B \times 3 = 5.72 \text{ MB}$	28.7

위와 같은 결과가 나타나는 이유를 분석해보면, 코드가 실행되는데 필요한 변수의 데이터 크기가 cache 크기보다 작을 경우 해당 데이터들을 cache에 옮기고 연산이 실행된다. 때문에 데이터 이동이 없어 연산에 필요한 클럭이 7 cycle로 일정하다. 연산에 필요한 변수의 데이터 크기가 cache 크기보다 크다면 cache에 변수를 넣지 않고 Fig. 2에서 확인할 수 있는 DDR memory 또는 CoreNet platform cache에서 연산에 필요한 변수를 가져와 연산을 수행한다. 그런데 데이터 이동속도가 CPU 연산 속도보다 느리기 때문에 데이터 이동이 보틀넥(bottleneck)이 되어 연산에 필요한 클럭 수가 증가하게 된다.

T2080 사용 설명 문서에 따르면 DDR에서 직접 변수값을 가져오는 경우 Core와 CoreNet coherency fabric 간의 bandwidth가 179 Gb/s(22 GB/s)로 제한된다[2]. 179 Gb/s로 대역폭은 CPU 1 cycle(1.8 GHz)

당 12.43 byte를 송수신할 수 있다. 이 점을 고려하여 Fig. 3와 같이 테스트 코드를 수정하여 사용 배열 크기를 조정했을 때는 Table 4의 결과 같이 급격한 성능저하가 발생하지 않았다.

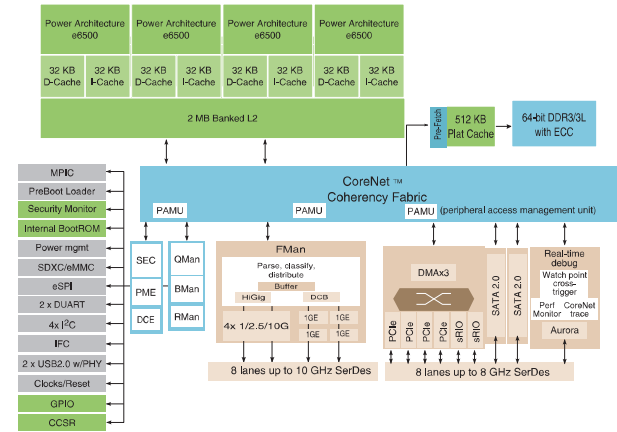


Fig. 2. T2080 block diagram[1]

```

array_size = 400000;

int *t_a = (int*)malloc(sizeof(int)*array_size);
int *t_b = (int*)malloc(sizeof(float)*array_size);
int sum = 0;

for(i = 0; i < array_size; i++){
    t_a[i] = (int)(rand()/1000);
    t_b[i] = (int)(rand()/1000)
}

_Vx_ticks_t startT, endT;

startT = tickGet();

for(j = 0; j < 2000; j++){
    for(i = 0; i < array_size; i++){
        sum = t_a[i] + t_b[i];
    }
}

endT = tickGet();
    
```

Fig. 3. Revised performance test code

Table 4. NXP T2080 revised performance test result

Data array	Data size	Calculation cycle
200,000	$200,000 \times 4B \times 2 = 1.53 \text{ MB}$	5.0
300,000	$300,000 \times 4B \times 2 = 2.59 \text{ MB}$	7.0
400,000	$400,000 \times 4B \times 2 = 3.05 \text{ MB}$	9.0

4. 추가시험 및 분석

PowerPC 및 Intel 계열의 다른 CPU에서도 동일한 결과가 나오는지 확인하기 위하여 PowerPC 계열의 P4080과 Intel 계열의 E3950으로 추가 시험을 수행하였다. 두 CPU의 스펙을 살펴보면, P4080은 메인 클럭 1.2 GHz, 코어 수 8개, 연산 성능은 30K DMIPS 이다. E3950은 메인 클럭 1.6 GHz, 코어 수 4개, 연산

성능은 42.1K DMIPS로, 코어 수 및 메인 클럭 속도를 고려하면 데이터 시트 기준 두 CPU의 연산 성능은 비슷하다. P4080의 경우 Fig. 4와 같이 T2080과 유사한 CoreNet coherency fabric 구조를 가진다.

앞선 시험과 동일하게 AltiVec 연산 가속 옵션은 사용하지 않고 데이터 크기를 변경하여 연산 소요 시간을 측정하였다. 수행 결과 Tables 5-6와 같이 P4080의 소요 시간이 더 길어 E3950의 연산 속도가 더 빨

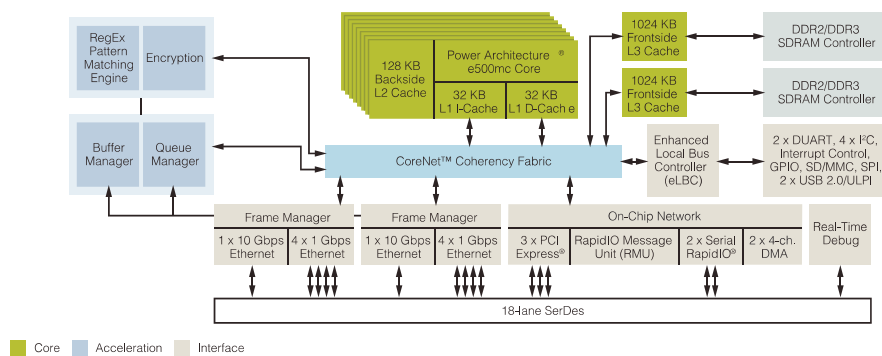


Fig. 4. P4080 block diagram[5]

Table 5. NXP P4080 performance test result

Data array	Integer calculation		Float calculation	
	Time (s)	Cycle	Time (s)	Cycle
5,000	0.020	4.8	0.040	9.6
6,000	0.040	8.0	0.050	10.0
7,000	0.040	6.9	0.060	10.3
8,000	0.050	7.5	0.070	10.5
9,000	0.050	6.7	0.080	10.7
10,000	0.060	7.2	0.090	10.8
12,000	0.090	9.0	0.130	13.0
14,000	0.130	11.1	0.170	14.6
16,000	0.160	12.0	0.210	15.8
18,000	0.200	13.3	0.240	16.0
20,000	0.220	13.2	0.280	16.8
30,000	0.360	14.4	0.440	17.6
40,000	0.500	15.0	0.610	18.3
50,000	0.620	14.9	0.760	18.2
60,000	0.750	15.0	0.920	18.4
70,000	0.870	14.9	1.070	18.3
80,000	1.000	15.0	1.220	18.3
90,000	1.240	16.5	1.500	20.0
100,000	1.590	19.1	1.870	22.4
150,000	2.580	20.6	2.940	23.5
200,000	3.540	21.2	4.010	24.1
400,000	7.350	22.1	8.380	25.1
600,000	11.020	22.0	12.540	25.1
800,000	14.400	21.6	16.510	24.8
1,000,000	21.690	26.0	24.140	29.0

Table 6. Intel E3950 performance test result

Data array	Integer calculation		Float calculation	
	Time (s)	Cycle	Time (s)	Cycle
10,000	0.065	10.4	0.066	10.6
20,000	0.166	13.3	0.163	13.0
30,000	0.249	13.3	0.244	13.0
40,000	0.261	10.4	0.260	10.4
50,000	0.329	10.5	0.328	10.5
60,000	0.391	10.4	0.390	10.4
70,000	0.460	10.5	0.460	10.5
80,000	0.522	10.4	0.519	10.4
90,000	0.618	11.0	0.617	11.0
100,000	0.864	13.8	0.849	13.6
150,000	1.051	11.2	1.052	11.2
200,000	1.722	13.8	1.688	13.5
400,000	3.436	13.7	3.362	13.4
600,000	5.188	13.8	5.086	13.6
800,000	6.889	13.8	6.751	13.5
1,000,000	8.630	13.8	8.442	13.5

랐다. 데이터 크기 변화에 따른 연산에 필요한 클럭 수 변화도 동일하게 P4080에서는 발생했고 Intel 계열인 E3950에서는 나타나지 않았다. 데이터 크기에 따른 필요 클럭 수를 그래프로 표현한 Fig. 6를 보면 E3950은 데이터 크기 변화에도 10~14 클럭을 유지하였다. 이에 반해 Fig. 5를 보면 P4080은 배열의 크기가 12,000과 90,000일 때 연산에 필요한 클럭 수가 상대적으로 크게 증가하였다. P4080은 T2080와 다르게 L2 cache를 공용으로 사용하지 않고 각 코어당 128 KB의 L2 cache를 배정하여 사용하고 필요에 따라 이를 다른 코어에 공유하여 사용한다. 이 때문에 데이터 크기가 L2 cache보다 커지는 배열 크기 12,000 이후부터 첫 번째 클럭 수 증가가 나타나고 이후 다른 코어의 L2 cache를 공용으로 사용하다 배열의 크기가 90,000보다 커질 때 데이터의 크기가 공용으로 사용할 수 있는 캐시의 최대 크기인 약 1 MB에 근접하고 이후 CCF 병목 현상이 나타난다고 유추할 수 있다. 따라서 앞의 시험과 동일하게 Intel 계열 CPU는 데이터 크기 변화에 필요한 클럭 수를 유지하였지만 PowerPC 계열은 그렇지 못했다.

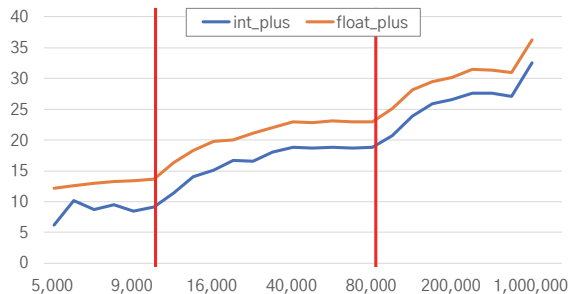


Fig. 5. NXP P4080 performance test result graph

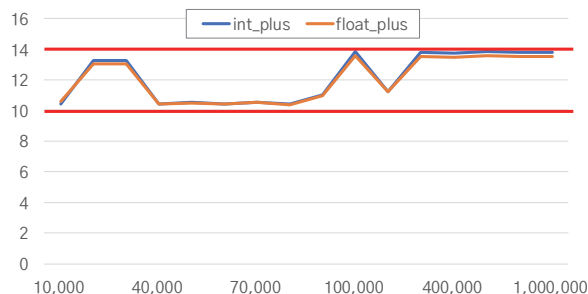


Fig. 6. Intel E3950 performance test result graph

5. 결론

데이터 시트의 GFLOPs 성능 지표로는 PowerPC 계열인 T2080이 Intel 계열 E3845보다 성능이 높았지만 AltiVec 가속 옵션을 사용하지 않은 조건의 성능 평가에서는 T2080의 성능이 더 낮았다.

또한 캐시 크기보다 큰 데이터에 대해서는 NXP에서 설계한 CCF 구조에서 병목 현상이 발생하여 연산 성능이 저하되는 경향을 보였다. 이에 반해 Intel 계열 CPU는 특정 데이터 크기에 대한 병목 현상을 보이지 않았고 데이터 크기가 커짐에도 연산에 필요한 클럭이 일정한 경향을 보였다.

따라서 영상 데이터 등 큰 데이터를 한 번에 연산할 경우 Intel 계열 CPU가 더 유리하다. AltiVec 가속 옵션을 적용할 경우 데이터 시트와 동일한 고성능의 연산을 사용할 수 있지만, 제공되는 library의 함수를 사용하기 위해서는 데이터 자료형 변환 등의 전처리 작업이 필요하고, 임베디드 환경 등 제한된 환경에서는 적용에 어려움이 있다. 영상 탐색기와 같이 영상 데이터를 처리하고 임베디드 환경에서 동작하는 시스템의 경우 Intel 계열 CPU를 사용하는 것이 시스템 구현에 더 유리할 것이라 판단된다.

참고문헌

- [1] Nxp Cor, "T2080 Product Brief," <https://www.nxp.com>
- [2] Nxp Cor, "A Deep Dive on the QorIQ T2080 Processor," <https://www.nxp.com>
- [3] Linley Gwennap. "AltiVec Vectorizes PowerPC," Microprocessor Report, 12(6), May 1998.
- [4] CPU-monkey, "Intel Atom E3845 Benchmark, Test and specs," https://www.cpu-monkey.com/en/cpu-intel_atom_e3845
- [5] Nxp Cor, "P4 Series P4080 multicore processor," <https://www.nxp.com>