

## Direct Memory Access untuk Menghitung Waktu Prosesor Intel Celeron N2840 dan AMD A8-7410 dalam Menangani Transfer Data

Abd. Charis Fauzan<sup>a,1,\*</sup>, Utrodus Said Al Baqi<sup>a,2</sup>, Asna Andi Auladi<sup>a,3</sup>, A. Zainul Muttaqin Sph<sup>a,4</sup>,

<sup>a</sup> Program Studi Ilmu Komputer, Universitas Nahdlatul Ulama Blitar, Indonesia

<sup>1</sup> abdcharis@unublitar.ac.id; <sup>2</sup> come.programmer@gmail.com; <sup>3</sup> asnaandi1108@gmail.com;

<sup>4</sup> azaqinakatsuki48@gmail.com

\* corresponding author

### ARTIKEL INFO

#### Article history

Diterima: 1 Juni 2019

Direvisi: 28 Juli 2019

Diterbitkan: 30 Agustus 2019

#### Keywords

Direct Memory Access

AMD A8 7410

Intel Celeron N2840

Transfer Data

### ABSTRAK

Penelitian ini bertujuan untuk mengetahui waktu yang dibutuhkan oleh prosesor Intel Celeron N2840 dan AMD A8-7410 dalam menangani transfer data dengan metode Direct Memory Access (DMA), sehingga dapat mengetahui prosesor yang lebih cepat dan efisien untuk digunakan dalam proses transfer data jika menggunakan DMA. Hasil perhitungan tersebut dapat menjadi indikator seberapa cepat atau lambat prosesor tersebut dalam menangani proses transfer data. Penelitian ini menggunakan 2 jenis prosesor, yang pertama yaitu Intel Celeron N2840 dengan kecepatan sebesar 2.16 GHz dan memiliki clockspeed sebesar 2583 MHz, sedangkan yang kedua yaitu AMD A8-7410 dengan kecepatan sebesar 2.20 GHz dan memiliki clockspeed sebesar 2395 MHz. Metode yang digunakan adalah direct memory access untuk menghitung waktu prosesor pada saat proses transfer data, beberapa data yang digunakan dalam perhitungan yaitu kecepatan transfer, *clockspeed*, *cyclus* dan *block*. Hasil perhitungan untuk prosesor Intel Celeron N2840 sebesar 20,90592334% waktu prosesor yang dibutuhkan pada saat transfer data dengan DMA, sedangkan untuk prosesor AMD A8-7410 sebesar 22,96450939 % waktu prosesor yang dibutuhkan pada saat transfer data dengan DMA. Dari hasil perhitungan waktu kedua prosesor tersebut dapat diambil kesimpulan bahwa prosesor Intel Celeron N2840 memiliki persentase lebih sedikit, sehingga dalam menangani transfer data, prosesor Intel Celeron N2840 lebih cepat dari pada prosesor AMD A8-7410 karena waktu yang digunakan dalam proses tersebut lebih sedikit.

### PENDAHULUAN

Direct memory access (DMA) ialah sebuah fitur dalam semua komputer modern saat ini yang memungkinkan perangkat untuk dapat memindahkan blok data yang besar tanpa interaksi dengan prosesor secara terus menerus. Sementara ketika perangkat mentransfer blok data, prosesor bebas untuk terus menjalankan intruksi lain tanpa khawatir tentang data yang ditransfer ke memori, atau ke perangkat lain. Ide dasarnya adalah kita dapat menjadwalkan perangkat direct memory access untuk melakukan tugasnya sendiri [1]. Hal tersebut dapat terjadi karena *direct memory access* memiliki mekanisme yang memungkinkan perangkat *I/O* untuk mengakses memori utama tanpa harus melalui *CPU* secara berulang-ulang, jadi dengan menggunakan *direct memory access* memungkinkan periferal atau perangkat tambahan membaca dari dan / atau menulis memori tanpa intervensi oleh *CPU* [2].

Fungsi utama dari DMA ialah sebagai modul tambahan (hardware) pada sistem bus dimana kontroler DMA mengambil alih tugas dari CPU untuk penanganan I/O. Pengontrol DMA dapat menghasilkan alamat memori dan memulai siklus membaca atau menulis memori. Hal itu mengandung beberapa register prosesor yang bisa dibaca dan ditulis oleh CPU. Ini termasuk alamat memori register, register penghitung byte, dan satu atau lebih

register kontrol. Register kontrol menentukan I / O port untuk digunakan, arah transfer (membaca dari perangkat I / O atau menulis ke perangkat I / O), dan unit transfer (byte pada suatu waktu atau kata pada suatu waktu) [3]. Penggunaan DMA didalam sebuah komputer modern memberikan keuntungan yaitu tingkat transfer data lebih cepat dan DMA sangat berguna jika sejumlah data yang besar perlu ditransfer. Untuk menentukan kecepatan transfer sebuah processor yang menggunakan DMA kita perlu menghitung *CPU time*. *CPU time* (waktu *CPU*) ialah jumlah persis total waktu yang dihabiskan *CPU* ketika memproses data untuk program atau proses tertentu, program dan aplikasi biasanya tidak menggunakan prosesor 100% dari waktu mereka berjalan, sebagian waktu itu dihabiskan untuk operasi *I/O* dan mengambil dan menyimpan data pada ram atau perangkat penyimpanan, waktu *CPU* berjalan hanya ketika program benar-benar menggunakan *CPU* untuk melakukan tugas-tugas seperti melakukan operasi penjumlahan, pengurangan, pembagian, perkalian dan logika, waktu *CPU* juga dikenal sebagai waktu pemrosesan [4].

Waktu *CPU* bisa diartikan sebagai pengukuran berapa lama waktu ketika data sedang dikerjakan oleh prosesor dan juga digunakan sebagai indikator seberapa banyak pemrosesan yang diperlukan untuk suatu proses atau seberapa intensif *CPU* pada suatu proses atau program [5]. Tergantung pada skenario penggunaan, penggunaan waktu *CPU* 100% untuk periode tertentu mungkin buruk atau baik. Ini bisa baik jika tujuannya adalah untuk menggunakan *CPU* sepenuhnya di antara program, atau dapat dikatakan buruk karena dapat menunjukkan bahwa suatu proses telah berjalan dalam pengulangan (loop) tanpa akhir atau bahwa proses tertentu tidak dioptimalkan [6]. Waktu *CPU* yang dibutuhkan oleh program dan proses seringkali sangat kecil, sepersekian detik, itulah sebabnya banyak program dapat berjalan bersamaan pada satu waktu yang sama, tetapi masih dapat mengaktifkan *CPU*. Masing-masing diberikan jumlah waktu atau irisan waktu tertentu untuk mengakses *CPU*. Misalnya, jika suatu program mengakses *CPU* selama satu detik setiap lima detik, maka total waktu *CPU* dalam rentang satu menit adalah 12 detik [7]. Namun pada saat ini, banyak pengguna komputer modern belum mengetahui jenis prosesor yang lebih efisien dalam melakukan proses transfer data. Sehingga banyak dari mereka kesulitan untuk mencari prosesor dengan kemampuan yang baik [8]. Didalam buku *Advanced Computer Architecture-CS501* telah dijelaskan sebuah contoh untuk menghitung waktu *cpu* (*the fraction of cpu time*) dalam menangani transfer data menggunakan *direct memory access* dengan sebuah prosesor yang memiliki clockspeed sebesar 500MHz dan pada sebuah *I/O device* dengan kecepatan transfer sebesar 10 MB/s diatas 100 Mb/s bus, data ditransfer pada blok 4 KB. Jika prosesor memiliki clockspeed sebesar 500 MHz maka ia memerlukan total waktu 5000 siklus untuk menangani setiap DMA request [9].

Ketika 2500 *DMA request* dikeluarkan untuk transfer data 4KB dengan kecepatan 10MB, total waktu yang diambil sebesar 12.500.000 siklus. Karena prosesornya memiliki clockspeed sebesar 500MHZ, fraksi waktu *CPU* yang dihabiskan adalah  $12.500.000 / (500 \times 10^6)$  atau 2,5%. Dengan menggunakan *direct memory access* kita dapat menghitung seberapa besar atau kecil waktu *CPU* pada saat menangani transfer data, sehingga dapat menjadi indikator seberapa cepat atau lambat *CPU* tersebut dalam menangani proses transfer data. Berdasarkan buku tersebut penelitian ini bertujuan untuk menghitung waktu *cpu* (*the fraction of cpu time*) Intel Celeron N2840 dan AMD A8-7410 dalam menangani transfer data dengan DMA, sehingga dapat mengetahui prosesor mana yang lebih cepat dalam proses transfer data menggunakan DMA.

## METODE

Waktu *CPU* (*The Fraction Of CPU Time*) maksudnya adalah jumlah persentase berapa lama *CPU* dalam menangani proses pengekseskuisian intruksi, untuk Menghitung fraksi waktu *CPU* (*CPU TIME*) membutuhkan beberapa data yaitu : kecepatan transfer,

*clockspeed*, *cyclus* dan *block*. Dari beberapa data tersebut kita dapat menghitung waktu *CPU* untuk menangani transfer data dengan rumus DMA :

$$FC = (v / b) * C / Vc$$

Dari rumus tersebut dapat diuraikan masing-masing sebagai berikut :

$FC$  = Fraction CPU time

$v$  = Kecepatan transfer

$Vc$  = *Clockspeed* (MHz)

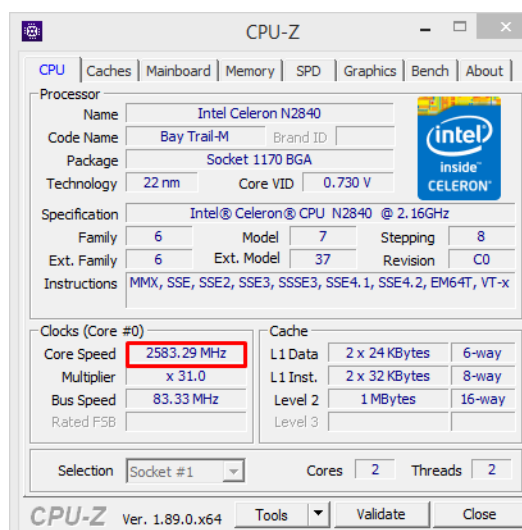
$C$  = *Cyclus* (cycle)

$b$  = *Block*

Kecepatan transfer yang dimaksud didalam rumus tersebut merupakan kecepatan dari perangkat I/O dalam mentransfer data, *Clockspeed* merupakan kecepatan dari CPU atau processor. *Cyclus* adalah waktu yang diperlukan untuk pelaksanaan satu operasi prosesor sederhana seperti perkalian, pengurangan, penambahan, perkalian, pembagian dan lain-lain atau bisa juga memiliki pengertian lain yaitu waktu yang digunakan untuk pengambilan (*fetching*) dan eksekusi perintah (*execution*) sebuah intruksi [10]. *Block* merupakan unit pada alokasi tempat penyimpanan dan juga transfer data biasanya berisi sejumlah seluruh catatan, memiliki panjang maksimum, dan ukuran blok.

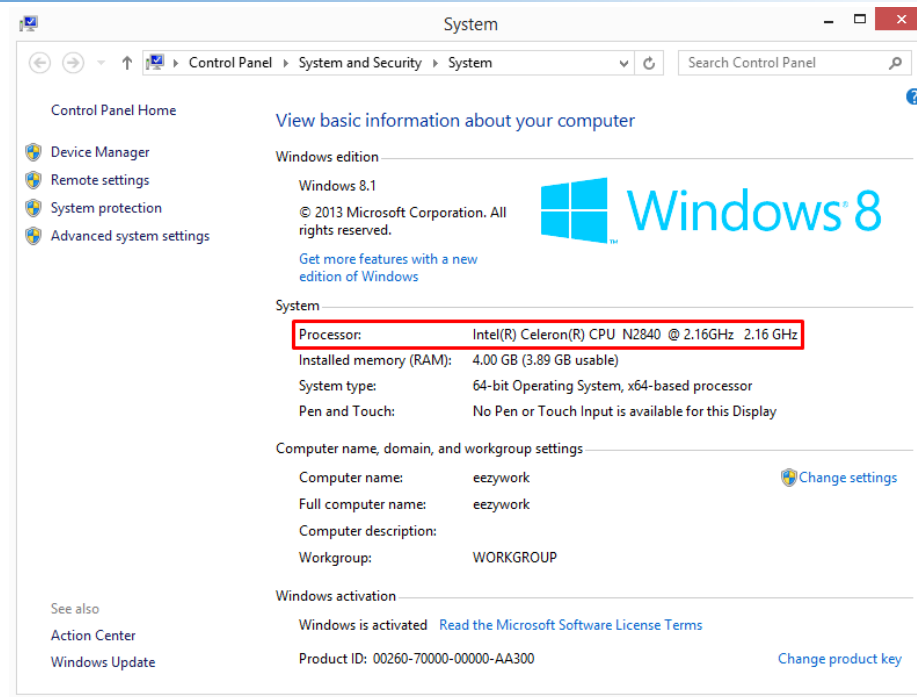
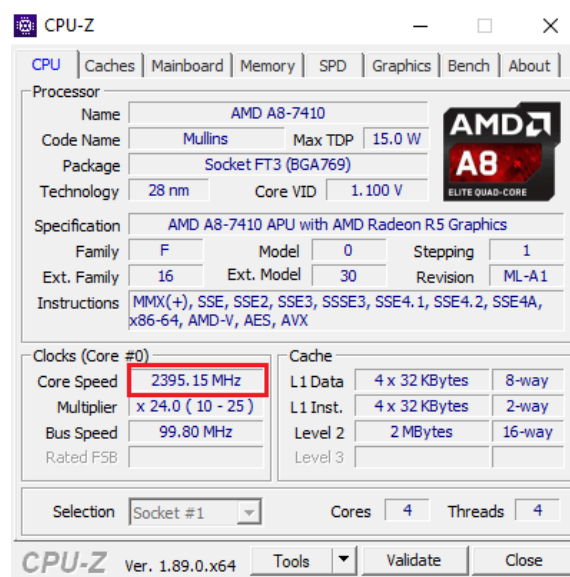
## PEMBAHASAN

Penelitian ini menghitung waktu *CPU* (*The Fraction of CPU Time*) dalam menangani transfer data dengan *DMA* pada sebuah I/O device yang memiliki kecepatan transfer sebesar 10 MB/s diatas 100 Mb/s bus, dimana data tersebut ditransfer pada blok 4 KB menggunakan 2 prosesor yang berbeda yaitu Intel Celeron N2840 dan AMD A8-7410. Spesifikasi masing-masing prosesor, di dapatkan dengan melihatnya menggunakan aplikasi *CPU-Z* dan juga informasi dasar dari 2 laptop yang berbeda, berikut ini adalah gambaran dari masing-masing *prosesor*:

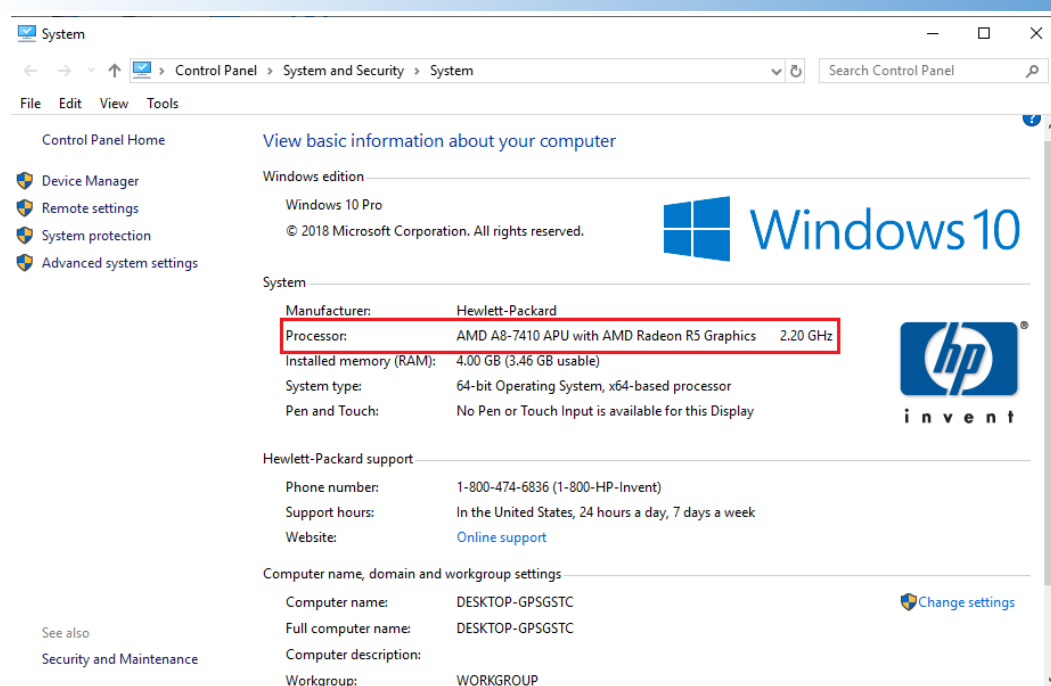


Gambar 1. *Clockspeed* dari Intel Celeron N2840

Gambar 1 menunjukkan tentang jumlah *clockspeed* dari prosesor Intel Celeron N2840 sebesar 2583.29 MHz yang di ukur menggunakan software *CPU-Z*. Sedangkan, Gambar 2 menunjukkan kecepatan yang dimiliki oleh prosesor Intel Celeron N2840 yaitu 2.16 GHz. Untuk melihatnya dengan cara masuk kedalam control panel selanjutnya masuk kedalam system and security dan yang terakhir klik system” sehingga terlihatlah sebagaimana Gambar 2.

Gambar 2. Kecepatan *processor* Intel Celeron N2840Gambar 3. *Clocks* dari AMD A8-7410

Gambar 3 menunjukkan tentang jumlah *clockspeed* dari processor AMD A8-7410 sebesar 2395.15 MHz yang di ukur menggunakan software CPU-Z. Sementara itu, Gambar 4 menunjukkan tentang jumlah *clockspeed* dari processor AMD A8-7410 sebesar 2395.15 MHz yang di ukur menggunakan software CPU-Z.

Gambar 4. Kecepatan *processor* AMD A8-7410

Tabel 1. Perbedaan Processor Intel Celeron N2840 dan AMD A8-7410

Nama Prosesor	Clockspeed	Kecepatan Processor	Cyclus
Intel Celeron N2840	2583 MHz	2.16 GHz = 2160 MHz	Cyclus = 2160 MHz * 10 = 21600 cycle
AMD A8-7410	2395 MHz	2.20 GHz = 2200 MHz	Cyclus = 2200 MHz * 10 = 22000 cycle

Tabel 1 menunjukkan bahwa perbedaan clockspeed kedua prosesor tersebut sebesar 2583 MHz dengan 2395 MHz, sedangkan untuk kecepatan kedua prosesor tersebut sebesar 2.16 GHz dengan 2.20 GHz. Untuk menemukan cyclus dari kedua prosesor tersebut dengan cara merubah kecepatan prosesor dari GHz menjadi Mhz, jadi kecepatan prosesor Intel Celeron N2840 sebesar 2160 MHz dan kecepatan prosesor AMD A8-7410 sebesar 2200 MHz. Selanjutnya dari kecepatan kedua prosesor yang telah diubah menjadi MHz dikalikan dengan 10 untuk menemukan berapa cycle dari kedua prosesor tersebut, sehingga cyclus untuk prosesor Intel Celeron N2840 sebesar 21600 cycle sedangkan prosesor AMD A8-7410 sebesar 22000 cycle. Dari data-data tersebut, dapat dihitung waktu CPU (*The Fraction Of Cpu Time*) dalam menangani transfer data menggunakan DMA.

Rumus perhitungan :

$$\text{Fraction CPU Time} = ((v / b) * C) / (Vc)$$

Berikut adalah hasil dari perhitungannya lakukan :

$$\begin{aligned} \text{Intel Celeron N2840} &= ((10/4) * 21600) / 2582 \\ &= 20,90592334 \% \end{aligned}$$

$$\begin{aligned} \text{AMD A8-7410} &= ((10/4) * 22000) / 2395 \\ &= 22,96450939 \% \end{aligned}$$

## KESIMPULAN

Dari hasil penelitian diatas, dapat diambil kesimpulan bahwa dari waktu CPU yang digunakan untuk menangani transfer data menggunakan prosesor Intel Celeron N2840 dan AMD A8-7410 hasilnya menunjukkan bahwa prosesor Intel Celeron N2840 memiliki persentase *Fraction of CPU time* lebih sedikit dan hal itu mempunyai arti bahwa, dalam hal

menangani transfer data, processor Intel Celeron N2840 lebih cepat dari pada menggunakan processor AMD A8-7410 karena waktu CPU yang digunakan dalam proses tersebut lebih sedikit. Sedangkan processor AMD A8-7410 menurut hasil perhitungan diatas menunjukkan persentase fraction of CPU time memiliki nilai yang lebih banyak sehingga waktu CPU yang digunakan dalam menangani transfer data menggunakan DMA lebih lambat daripada Intel Celeron N2840.

#### REFERENSI

- [1] V. S. Vanita, "Direct Memory Access ( DMA ) Controller," *INTERNATONAL J. Innov. Res. Technol.*, vol. 1, no. June, pp. 1069–1071, 2004.
- [2] A. Chandan, Kumar. Prajapati, Meenakshi. Verma, "Direct Memory Access," *Int. J. Innov. Res. Technol.*, vol. 1, no. 7, pp. 347–350, 2014.
- [3] A. Aljumah and M. Altaf, "Amba Based Advanced DMA Controller for SoC," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 3, pp. 188–193, 2016.
- [4] Y. Chi and Z. Zheng, "A Design of Direct Memory Access Controller for Wireless Communication SoC in Power Grid," *2018 IEEE 2nd Int. Conf. Circuits, Syst. Simulation, ICCSS 2018*, pp. 76–79, 2018.
- [5] W. A. Halang, "On methods for direct memory access without cycle stealing," *Microprocess. Microprogramming*, vol. 17, no. 5, pp. 277–283, 1986.
- [6] J. D. McCalpin, "Memory Bandwidth and Machine Balance in Current High Performance Computers," *IEEE Tech. Comm. Comp. Arch*, no. December 1995, 1995.
- [7] W. Stallings, *COMPUTER ORGANIZATION AND ARCHITECTURE DESIGNING FOR PERFORMANCE EIGHTH EDITION, EIGHT*. Upper Saddle River, New Jersey: Pearson Education, Inc, 2007.
- [8] S. Choirunnisa, R. Sarno, and A. C. Fauzan, "Optimization of Forecasted Port Container Terminal Performance Using Goal Programming," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 332–336.
- [9] H. F. Heuring, Vincent P. Jordan, *Advanced Computer Architecture-CS501, Lecture Handouts*. .
- [10] J. Xamán *et al.*, "Evaluation of the CPU time for solving the radiative transfer equation with high-order resolution schemes applying the normalized weighting-factor method," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 208, pp. 45–63, 2018.