

Feasibility Study on the Use of Low-Power Mini PC for Magnetic Resonance Imaging Application

Adnan^{1,a)}, Intan Sari Areni^{2,b)}, Zulkifli Tahir^{1,c)}, Bayazid^{1,d)}, and Riswandi^{1,e)}

¹*Department of Informatics, Hasanuddin University, Makassar, Indonesia*

²*Department of Electrical Engineering, Hasanuddin University, Makassar Indonesia
Jalan Perintis Kemerdekaan km. 10, Tamalanrea, Makassar, Indonesia*

^{a)}adnan@unhas.ac.id

^{b)}Corresponding author: intan@unhas.ac.id

^{c)}zulkifli@unhas.ac.id

^{d)}bayazidsustamy@gmail.com

^{e)}riswandir19d@student.unhas.ac.id

Abstract. Digital medical imaging is a process of constructing digital images for medical applications. One of them is Magnetic Resonance Imaging or MRI. Simple image construction uses the Inverse FFT algorithm which is an efficient algorithm. This study is to evaluate the performance of a magnetic resonance imaging system using a mini-computer system with low power. The design constraints are the system must be energy-efficient and also low cost. The first result of this study indicates the mini PC jetson Nano can reconstruct MRI images in a maximum time of 67.13 seconds from k-space data using the inverse FFT algorithm measuring 2048 with an electrical energy consumption of 150 watts-second (Joule). The second result of this research shows that the MRI reconstruction software makes the CPU spends time the most at the DoRSS function. The latest result of this research inspired us to do new research to improve performance and to reduce energy consumption further.

INTRODUCTION

MRI is now commonly used by health professionals to non-invasively detect organ abnormalities in the human body. Using this equipment, the patient lies on a flatbed. This flatbed can be slid into a large tube inducing a very strong magnetic field (0.5 - 1.5 T) [1] and a radio wavefield with a certain frequency and pattern to obtain several pieces of the organ's image called slices. Slices are stored as data in a discrete Fourier domain called k-space on computer storage. The process of getting k-space data is called the scanning process. Furthermore, to be able to see the slices organ image, the k-space data is processed using an inverse discrete Fourier transform algorithm using a computer. A more efficient Inverse Fourier Transform computation can be performed with a Fast Fourier algorithm provided that the size of the input data is a multiple of 2 to the power of N integers. To improve the obtained image, pre-processing can be done such as filtering and removing image artifacts. MRI machines consumed electrical energy a lot. In total, the MRI operation consumes a huge amount of electricity ranging from 9 kW to 18.4 kW based on a report [2]. In another article [3], MRI and also CT machine consumes electrical energy as much as energy consumption of a town of 854 people. During operational hours, the MRI scanner subsystem cannot just merely be turned off, even though it is to save electricity consumption. MRI scanners can only be made in a low-power state. However, this does not mean that there is no opportunity to reduce energy consumption associated with MRI operations. Saving electrical energy can be done on supporting devices such as acquisition controllers, image reconstruction functions, image storage, and retrieval, viewing, and post-processing.

In this paper, we present a performance evaluation on the Jetson Nano Single Board Computer in the MRI Image Reconstruction. The Jetson Nano performs Inverse FFT to reconstruct the MRI image. The contributions of this paper are :

1. Execution time and energy consumption increase as faster as the Inverse FFT size.
2. With 2048 of inverse FFT size, Jetson Nano consumes electrical energy as low as 150 watt-seconds.
3. The hotspot of the MRI reconstruction algorithm is located on a function called DoRSS.

RELATED WORKS

Several works of telehealth projects employing embedded systems such as mini PC and microcontrollers. The most popular work is the GNU Health embedded sub-project. GNU Health combines Social Medicine with the latest advances in bioinformatics, providing a framework for integrative medicine, governments, and Public Health institutions as well as research organizations [4]. GNU Health comprises several sub-projects such as Hospital Management Information System, Laboratory Information System, Thalamus, and GNU Health Embedded which built with Raspberry Pi.

Another work that employs Raspberry Pi is OrthancPi [5]. The OrthancPi is a DICOM PACS mini server ported from a project called Orthanc. The OrthancPi store and communicate medical Images (CT, MRI) from a server to one or more clients. This works shows that the first generation of Raspberry PI is capable of serving medical Image storage and communication. Similar to the OrthancPi, PumpkinPi [6] is a DICOM viewer using Raspberry Pi. This research is the extension of the student's final project [7]. The final project employs CUDA GPU on a laptop PC whereas this research employs Jetson Nano mini PC. Although the mini PC has CUDA GPU, we port the program such that it may run on a wide spectrum of mini PC without all about CUDA.

MATERIAL AND METHODOLOGY

Raspberry Pi 4b and Jetson Nano Mini PC Comparison

The selection of computer hardware to reconstruct magnetic resonance imaging images needs to consider the performance aspects of response time and electrical energy consumption. In this case, the response time performance can be used to determine whether a computer hardware platform is suitable to serve the processing of some scanned data within a critical time limit. Whereas the performance of electricity consumption can be used to assess the feasibility of operational costs for reconstructing medical images operating from morning to evening for 10 hours.

In terms of processing time, processors made by Intel are well known and dominate laptops, desktop PCs, and servers. But these devices are also known to be very wasteful in terms of electricity consumption. Conversely, in terms of low power consumption, ARM-made processors have long been known. That is why their processors dominate the portable device market such as smartphones and tablet PCs. But behind the low power consumption, the processing performance of ARM processors is not comparable to processors made by Intel. For this reason, our research uses ARM-based mini PCs.

There are several popular low-power mini PCs based on ARM processors. The first is Raspberry PI 4B and the second is Nvidia Jetson Nano. Table 1 shows Raspberry Pi 4B and Jetson Nano comparison. The Raspberry PI 4B uses a 1.5GHz Quad-Core ARM A72 [8] processor while the Jetson Nano uses a 1.43 GHz Quad-Core ARM A57 processor [9]. Comparing the two mini PC processors it looks like the Raspberry PI 4B is superior to the Jetson Nano. Raspberry PI 4B also offers an alternative amount of RAM 8GB greater than Jetson Nano with 4GB RAM. The existence of the Nvidia Tegra GPU can provide additional computing power that is more powerful if programmed with CUDA. But this additional computing power must be paid for with more power consumption and more expensive prices. Figure 1.a shows a form factor of Raspberry Pi 4B, while Fig. 1.b. shows a form factor of Jetson Nano.

Both Raspberry PI 4B mini PCs and Jetson Nano use micro-SD media as boot media and main storage media. The availability of USB 3.0 facilitates larger capacity and faster storage media with interfaces such as NVME to USB 3.0 adapters.

TABLE 1. Raspberry Pi 4B and Jetson Nano comparison

	Raspberry Pi 4B	NVIDIA Jetson Nano
CPU	Quad Core ARM A72 @1.5 GHz	Quad-Core ARM A57 @1.43 GHz
GPU	Video core VI	NVIDIA Tegra 128 Core Maxwell
Main Memory	1 GB, 2 GB, 4GB, 8 GB LPDDR4	4 GB LPDDR4, 25.6 GB/s
Storage	Micro-SD	Micro-SD
Connectivity	Gbe, 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE	Gigabit Ethernet, M.2 Key E slot
Display	2x micro HDMI	HDMI and DP
USB		4x USB 3.0, USB 2.0 micro-B
Mechanical Dimension	86 mm x 56 mm	100 mm x 80 mm
Estimated CPU Performance	4.72 DMIPS/core MHz	4.6 DMIPS/core MHz



(a)



(b)

FIGURE 1. Single Board Computers. (a). Raspberry Pi Model 4B. (b). Jetson Nano is the bigger one

Material Hardware

This research is to make use of Jetson Nano mini PC in MRI image reconstruction. The right column of Table 1 presents the specification of the mini PC. In addition to micro SD as main storage, we equipped the Jetson Nano with NVME Solid State Drive. The SSD is attached on NVME PCIe to USB 3.0 adapter and connected to Jetson Nano's USB 3.0 port. The SSD is to store k-space data in two separate files namely the hdr file and cf file. The Jetson Nano mini PC is powered by a 5V DC power supply capable of delivering a maximum 3A electrical current. It is operated through direct interaction using a keyboard, mouse, and LCD. In summary, Table 2 lists all the parts used in this experiment.

TABLE 2. Parts list

Devices	Specification
Jetson Nano	As presented in table 1 column 3th
Micro-SD main storage	16 GB UHS-1 Class 10
SSD	NVME SSD PCIe M2 B-Key 2242, 512 GB Write speed: 436.9 MB/s Read speed: 450 MB/s
User Interface devices	USB keyboard, USB mouse and 20" LCD
NVME to USB 3.0 adapter	6 gbps
DC Power Supply	5 V 4A with DC jack

Software

Some tools are required to build and evaluate MRI image reconstruction software. The tools are listed as follows

- g++ 7.40 compiler
- OpenCV 3.3.1
- FFTW 3.3 (fast Fourier transform library)
- GNU gprof, (program execution profiler)
- Tegrastat

Experimental Configuration

The first step of this research is to implement software for MRI reconstruction. For its method is the inverse FFT algorithm. The sizes of Fourier transform matrices are 512, 1024, 2048 respectively. When data is not aligned with the size, zeroes are filled into the matrices.

Some experiments are conducted to evaluate the performance of the software on the Jetson Nano mini PC. The first performance evaluation is to measure execution time. to measure the execution time of the MRI reconstruction process is simple. We measure the duration between program start time and end time. That is done by calling the clock function that is available on many C compilers.

The second experiment is to find the locations of the program where the execution takes time the longest. This is an experiment to be called execution profiling because it produces a program execution profile. The produced execution profile is useful to optimize program execution time.

The last evaluations are to estimate the power consumption of mini pc while running the MRI reconstruction process. The estimation of power consumption is easy to do. In this evaluation, Jetson Nano software tools include Tegrastat. The tegrastat can estimate the power consumption of the Jetson Nano while running an image reconstruction process.

Both execution time and power consumption are then to calculate energy which the jetson Nano consumed to reconstruct MRI images.

EXPERIMENTAL RESULTS

The result of this research is a program that can reconstruct images from k-space using a low-power mini PC, i.e Jetson Nano. In this work, the program is made to run serially on its CPU of Jetson Nano. Hence produced program is a sequential program.

The figure shows the images reconstructed by the program using the ARM Cortex A57 CPU and 4 GB RAM. The image is a knee image that was reconstructed from k-space data using a 2D inverse fast Fourier transform of size 1024.



FIGURE 2. MRI Image reconstructed through inverse FFT with Zero Filling. (a) Image by 512x512 inverse FFT, (b) Image by 1024x1024 inverse FFT, (c) Image by 2048x2048 Inverse FFT

Execution Time

First of all, this subsection is to discuss the execution time for the MRI image reconstruction program. In Fig. 3, the presented data make up a function over the size and the number of slices of the image in the Fourier transform domain. These experiments vary the size of the 2D Fourier transform from 512 to 2048. These also vary the number of slices of the image being reconstructed. As the number of slices doubles, the execution time also doubles. The rate of execution time increase as the 2D Fourier size increases faster than the number of slices. As the 2D Fourier size doubles, its corresponding execution time triples. The longest execution time is 67.13 seconds. This result of reconstruction time suggests that Jetson Nano may produce 53 of 4 slices MRI images in one hour.

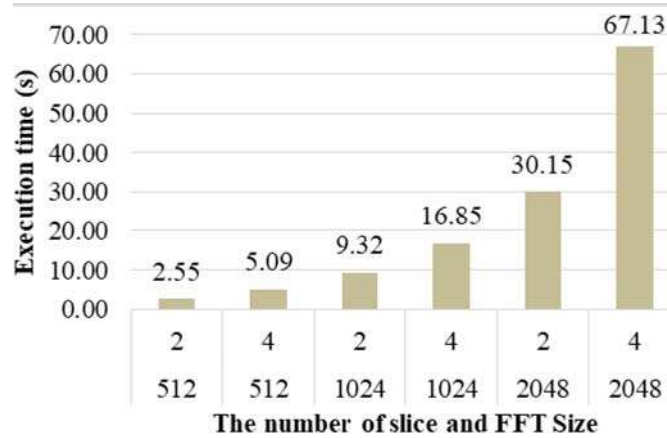


FIGURE 3. Program execution time

Execution Profile

Profiling execution of a program produces data that could suggest to us where the CPU(s) spent time the most. Knowing where is the hotspot location in a program useful in a process of program cost optimization. Figure 4 tells where the hotspot of program execution is. The MRI image reconstruction program spent time the most at a function called *DoRSS*. The function is to transform the program from a complex array into a magnitude array. In the *DoRSS* function, there are 4 nested loops. That's the way the time of *DoRSS* grows exponentially.



FIGURE 4. Program Execution profile. The hotspot of the MRI Image Reconstruction energy consumption is in *DoRSS* Function

Figure 5 shows electrical energy consumption as the slice number and FFT dimension changed. The energy is calculated from average power consumption (2.234 watts) and program running time (2.55 sec – 67.13 sec). From the chart, the dimension of the FFT algorithm increases the consumption of electrical energy faster than the number of slices. The larger the matrix size and the number of slices, the longer the processing time and the greater the electrical energy used.

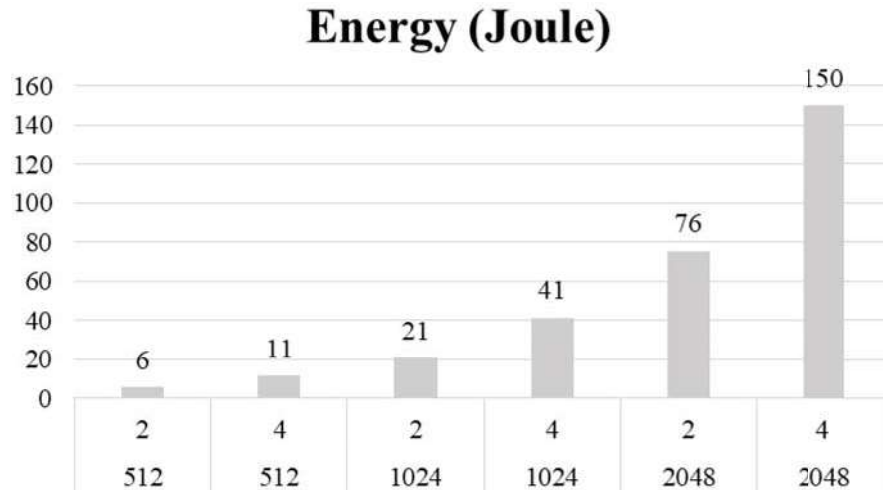


FIGURE 5. Energy consumption of the Jetson Nano mini PC while running MRI Image Reconstruction

CONCLUSION

In this research, we studied the feasibility of a mini PC, i.e. Jetson Nano as a computing device in MRI Image reconstruction. We demonstrated that the mini PC can reconstruct MRI images within an acceptable time interval and at very low power consumption. The image reconstruction from k-space data using a 2D inverse FFT (with the size of 2048) takes time 67.13 seconds and consumes energy as much as 150 watt-second (joule).

REFERENCES

1. A. Murphy and J.R. Ballinger, Magnets (types), <https://radiopaedia.org/articles/magnets-types> (accessed: Apr. 1, 2020).
2. J.J. Knott, L. Varangu, K. Waddington, T. Easty, and S. Shi, *Medical Imaging Equipment Energy Use Study* (The Canadian Coalition for Green Health Care, 2017).
3. T. Heye, R. Knoerl, T. Wehrle, D. Mangold, A. Cerminara, M. Loser, M. Plumeyer, M. Degen, R. Lüthy, D. Brodbeck, and E. Merkle, *Radiology* **295**, 593–605 (2020).
4. Martin and F. Luis, *GNU Health: A Free/Libre Community-Based Health Information System* (Association for Computing Machinery, Berlin, 2016).
5. M. Barnig, Orthanc Raspberry Pi, <http://www.web3.lu/orthanc-raspberry-pi/> (accessed: Mar. 15, 2016).
6. S. Doyle, Pumpkin PACS/Pumpkin Pi, <https://projects-raspberry.com/pumpkin-pacspumpkin-pi/> (accessed: Nov. 15, 2020).
7. M.R. Hatsa, Zero Filling MRI Medical Image Reconstruction Software, Universitas Hasanuddin, 2017.
8. Raspberry Pi Foundation, Raspberry Pi 4 Tech Speed, <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/> (accessed: May 1, 2020).
9. NVIDIA, Jetson Nano, <https://developer.nvidia.com/embedded/jetson-nano> (accessed: Aug. 5, 2020).