Research on improving the performance of telecommunication broadcasting and warning systems from the type of microprocessor platforms

A A Kolpakov, A A Belov, Y A Kropotov
Murom Institute (branch) "Vladimir State University named after Alexander and Nicholay Stoletosv", Orlovskaya street 23, Murom, Russia, 602264

Abstract. The article discusses the use of various microprocessor platforms for the development of terminal devices of telecommunication broadcasting and warning systems. Their characteristics are investigated and the performance of the selected platforms is tested when mixing audio streams. The possibility of using parallel audio processing algorithms is also being considered.

1. Introduction
One of the important issues of developing telecommunication systems is the choice of a hardware platform. It is important that the selected platform is consistent with the given parameters, because it is very difficult to change when the software has already been developed.

Development of a modern telecommunication broadcasting and notification system involves a high-performance microprocessor platform with the specified technical parameters and appropriate software [1]. It is also possible to use parallel computing algorithms for increasing the performance of a telecommunication system, therefore, the selected microprocessor platform should be able to implement the above algorithms.

2. Formulation of the problem
SK-iMX6ULL was chosen as the main microprocessor platform. It is produced by the domestic company Starterkit based on the NXP iMX6ULL processor. This platform uses a single-core ARM Cortex-A7 800 MHz processor, has 512 MB DDR3 RAM, 8 GB eMMC Flash memory, an Ethernet 100 / 10M PHY network adapter, and the necessary connectors for sensors. [2] The Linux operating system is used. This single-board microcomputer was chosen due to the fact that it has built-in memory and has the ability to play sound through the built-in sound card. The computational capabilities of the processor are sufficient for processing data from sensors and transmitting them over the local network, as well as for reproducing sound fragments received over the network or stored in local memory. At the same time, these capabilities are completely insufficient for mixing a large number of audio streams and video playback. Therefore, when expanding the functionality, a more productive platform was required.
Figure 1. The layout of the broadcast and notification device based on SK-iMX6ULL.

For microprocessor stream interception devices required more productive microprocessor system. In this case, the possibility of using single-board computers Raspberry PI3 B + and Orange PI One is considered.

The Raspberry PI3 B + is based on the Broadcom BCM2837B0 processor, which has 4 64-bit Cortex-A53 cores with a clock frequency of 1.4 GHz, 1 GB of RAM, a Gigabit Ethernet network adapter and a GPIO connector for connecting peripherals. [3] It uses the Linux operating system based on Debian. A single-board computer is shown in figure 2.

Figure 2. Raspberry PI3 B + Single Board Computer.

Orange PI One is a cheaper simplified copy of Raspberry PI, built on the Allwinner H3 platform, has a 32-bit 4-core ARM Cortex-A7 processor with a clock frequency of 1.2 GHz, 512 MB RAM, supports data transfer over a local network via Ethernet, as well as a GPIO connector, unified with Raspberry PI. [4] It use the Linux operating system, similar to Raspberry PI3 B +. A single-board computer is shown in figure 3.
As can be seen from the characteristics, both platforms have the ability to multithreaded data processing due to multi-core processors, and also have sufficient performance to play the video stream and mix audio streams in real time. However, they have some disadvantages. Both platforms require a microSD card to load the operating system and do not have internal memory. Also in Orange PI One, RAM is clearly not enough to process large amounts of information. The cost of the Raspberry PI3 B + is slightly higher, but it has developed technical support and a large community of developers.

All of the single-board computers presented above run on the Linux operating system, but the last two have a much larger repository for software components, which greatly simplifies software development. However, to use parallel data processing with a large number of threads, a microprocessor system with a large number of computing elements is required. Therefore, it was decided to pay attention to the NVIDIA Jetson Nano single-board computer, which is shown in figure 4.

This microprocessor system is based on the quad-core ARM Cortex-A57 processor, has an integrated NVIDIA Maxwell graphics processor with 128 NVIDIA CUDA cores, 4 GB of RAM and 16 GB of constant memory. [5] There is also Gigabit Ethernet support and peripheral connectors, including an SDIO compatible Raspberry PI3 B + connector. Because this microprocessor system has
support for NVIDIA CUDA technology, it can be involved in parallel processing of audio data with a large number of streams. However, first you need to compare the performance of all platforms on similar tests.

3. Comparative testing of microprocessor systems of data processing

Further development of the telecommunication broadcasting and notification system is a telephone and conference communication system, which requires real-time mixing of audio streams. It is also planned to finalize the software for organizing video communications. Because the developed system is a peer to peer and there is no dedicated server, which could take on the burden of mixing flows, the microprocessor-based terminal must have sufficient performance.

Having examined the characteristics of the above single-board computers, we can say with confidence that the SK-iMX6ULL does not have the necessary characteristics, while the other three systems can be used as the basis for a telephone and conference communication terminal using the appropriate peripherals and software. However, a more detailed comparison of their characteristics and the corresponding tests are required.

Table 1 shows the comparative characteristics of all the microprocessor systems presented.

Table 1. Comparative characteristics of the microprocessor systems under consideration

| Parameters       | Specification          | SK-iMX6ULL | Raspberry Pi3 B+ | Orange Pi One | NVIDIA Jetson Nano |
|------------------|------------------------|------------|------------------|---------------|--------------------|
| CPU cores        |                        | 1          | 4                | 4             | 4                  |
| CPU Frequency    |                        | 800 MHz    | 1400 MHz         | 1200 MHz      | 1500 MHz           |
| CPU Architecture| ARM Cortex-A7          |            | ARM Cortex-A53   | ARM Cortex-A7 |                   |
| Graphic Core     | VideoCore IV           |            | Mali400MP2       |               | NVIDIA Maxwell     |
| RAM              | 512 Mb                 |            | 1024 Mb          | 512 Mb        | 4096 Mb            |
| Flash            | 8 Gb                   |            | MicroSD          | MicroSD       | 16 Gb              |
| Lan              | 1000                   |            | 10000            | 100           | 10000              |
| OS               | Linux                  |            | Linux            | Linux         | Linux              |
| Price            | 3000 rub.              |            | 5000 rub.        | 1800 rub.     | 10000 rub.         |

As you can see from the table 1, NVIDIA Jetson Nano has the highest price, but at the same time it has the highest characteristics. Its advantage is the latest version of the processor core, the largest amount of RAM and the availability of permanent Flash storage. At the same time, although three of the above devices include a graphics processor, only NVIDIA Jetson Nano has the ability to use it as a computing module. Although more powerful microprocessor systems exist on the market, the presence of CUDA cores makes NVIDIA Jetson Nano very attractive for parallel algorithms. However, it is nevertheless worth evaluating its performance relative to the other listed devices.

The first test that was conducted was the Pi number calculation program, the source code of which is given below.

```python
import time

def make_pi():
    q, r, t, k, m, x = 1, 0, 1, 1, 3, 3
    for j in range(10000):
        if 4 * q + r - t < m * t:
            yield m
            q, r, t, k, m, x = 10*q, 10*(r-m*t), t, k, (10*(3*q+r))/t - 10*m, x
        else:
            q, r, t, k, m, x = q*k, (2*q+r)*x, t*x, k+1, (q*(7*k+2)+r*x)/(t*x), x+2
    t1 = time.time()
```
pi_array = []
for i in make_pi():
    pi_array.append(str(i))
pi_array = pi_array[:1] + ['.'] + pi_array[1:]
pi_array_str = ''.join(pi_array)
print("PI:", pi_array_str)
print("dT:", time.time() - t1)
The test results are presented in the table below.

Table 2. Experimental testing of microprocessor systems presented. Pi calculation

| Microprocessor system | Time, sec |
|-----------------------|-----------|
| SK-iMX6ULL            | 10.6      |
| Raspberry PI3 B+      | 3.06      |
| Orange PI One         | 4.1       |
| NVIDIA Jetson Nano    | 0.8       |

As you can see from the test results presented above, even without parallel computing and using CUDA cores, the NVIDIA Jetson Nano device shows much higher performance.

It is further proposed to conduct a test using parallel computing. [6,7] As can be seen from the characteristics of the systems under study, the use of parallel systems in SK-iMX6ULL is impossible, therefore this device will not be investigated in the next test.

The testing program counts the sines in parallel. Its source code is given below.

```python
import pycuda.driver as drv
import pycuda.autoinit
from pycuda.compiler import SourceModule
import numpy
import time

blocks = 64
block_size = 128
nbr_values = blocks * block_size
n_iter = 100000
print("Calculating %d iterations" % (n_iter))
print()

#########
# SourceModule SECTION

start = drv.Event()
end = drv.Event()
mod = SourceModule("""
    __global__ void gpusin(float *dest, float *a, int n_iter) {
        const int i = blockDim.x*blockIdx.x + threadIdx.x;
        for(int n = 0; n < n_iter; n++) {
            a[i] = sin(a[i]);
        }
        dest[i] = a[i];
    }
""")
gpusin = mod.get_function("gpusin")
a = numpy.ones(nbr_values).astype(numpy.float32)
dest = numpy.zeros_like(a)
start.record() # start timing
gpusin(drv.Out(dest), drv.In(a), numpy.int32(n_iter), grid=(blocks,1), block=(block_size,1,1))
```
end.record()  # end timing
end.synchronize()
secs = start.time_till(end)*1e-3
print("PyCUDA time and first three results:")
print("%fs, %s" % (secs, str(dest[:3])))
print()
# CPU SECTION
a = numpy.ones(nbr_values).astype(numpy.float32)
t1 = time.time()
for i in range(n_iter):
a = numpy.sin(a)

print("CPU time and first three results:")
print("%fs, %s" % (time.time() - t1, str(a[:3])))

This program uses the capabilities of NVIDIA CUDA [8, 9], therefore, calculations were performed on both the CPU and the GPU for the NVIDIA Jetson Nano device. The test results are shown in the table below.

| Microprocessor system     | Time, sec |
|---------------------------|-----------|
| Raspberry Pi3 B+          | 41.85     |
| Orange Pi One             | 58.6      |
| NVIDIA Jetson Nano CPU   | 13.3      |
| NVIDIA Jetson Nano GPU   | 0.67      |

Just like in the first test, the study showed that the performance of the NVIDIA Jetson Nano CPU is much higher than other available devices. However, in this test, the advantage of parallel data processing using NVIDIA CUDA is clearly visible.

The above tests showed that NVIDIA Jetson Nano is superior in performance to previously used microprocessor systems and can be successfully used as the basis for telephone and conference communications using parallel audio processing algorithms.

4. Experimental application of the audio stream mixing algorithm on selected microprocessor systems

The use of a microprocessor system as the basis for a telecommunication device for telephone and conference calls involves the use of a mixing algorithm for audio streams from abonents. Because SK-iMX6ULL platform showed insufficient performance in previous tests, it was decided not to investigate it in the future.

A two-part testing software package was developed [11]. The software for a personal computer generates audio streams for subscribers and broadcast them to a local network. Software for a microprocessor system should receive network packets with audio samples, unpack them, mix and output to an audio device. UDP was chosen as the basis for the data transfer protocol, the audio stream was encoded by the G.711 codec, and the stream was transmitted 256 kbps. Packet encryption was not performed.

Audio data processing was organized using a previously developed algorithm [10] using the multicore. This algorithm was also modified to use the NVIDIA CUDA technology. [12-14] The test results are shown in figure 5.
Figure 5. The results of experimental testing of the algorithm for mixing audio streams on the studied platforms.

As you can see from the test results, up to 10 abonents are mixed with an acceptable delay of about 2 second, although it is clear that the more modern architecture of the NVIDIA Jetson Nano microprocessor allows you to get some advantage. However, an increase in the number of abonents leads to the appearance of a playback delay of more than 3 seconds, which is unacceptable in the developed device. To eliminate this delay, it is possible to optimize the algorithm used, however, it should be borne in mind that in this mode the microprocessor, in addition to processing audio streams, also processes requests from the operating system, processes network packets, and decodes audio samples. Transferring only the audio mixing algorithm to a separate NVIDIA Maxwell computer [15-16] allows you to get a very significant performance increase: the delay remains within 1 second throughout testing. Thus, we can conclude that the use of the NVIDIA Jetson Nano platform is fully justified.

5. Conclusions
An experimental study showed that the use of the NVIDIA Jetson Nano platform in combination with calculations using the NVIDIA Maxwell GPU allows to obtain a delay in mixing audio streams from 20 subscribers in 0.22 seconds, while using a modern CPU ARM Cortex-A57 gives a delay of 7.1 sec., and the use of the previous generation CPU - in 18.61 sec. It is possible to further increase productivity by using parallel algorithms for other tasks of a telecommunication system. It is also obvious that the use of a high-performance platform allows you to apply better parameters of the audio stream, as well as enable encryption.

The NVIDIA Jetson Nano platform has a GPIO connector unified with the Raspberry PI, which allows you to connect already developed peripherals, has a large number of peripheral connectors, and can also be presented as a separate processor module for which you can develop your own platform using the necessary peripherals. The disadvantage of this platform is the high cost, so its use is advisable in the case of high-performance computing with widespread use of parallel algorithms.

Acknowledgments
The work was carried out under a grant from the president of the Russian Federation for state support of young Russian scientists № МК-2378.2020.9.
References

[1] L. Huo G Yang Y Liu and S Yao 2020 Design and Implementation of Universal Platform for Teaching Experimental of Microcomputer Relay Protection ACM International Conference Proceeding Series, 2020 International Conference on Computers, Information Processing and Advanced Education, CIPAE 2020; Ottawa pp. 306-311

[2] Manual SK-iMX6ULL [Electronic resource]. – Access mode: http://www.starterkit.ru/html/doc/Manual-SK-iMX6ULL.pdf (13.09.2020).

[3] Raspberry Pi Documentation [Electronic resource]. – Access mode: https://www.raspberrypi.org/documentation/ (13.09.2020).

[4] Allwinner H3 datasheet v.1.2 [Electronic resource]. – Access mode: https://drive.google.com/drive/folders/1JmgtWTIGA-hPv47cLtEYZa-Y3UZPSQNN (13.09.2020).

[5] NVIDIA Jetson Nano_Developer Kit User Guide [Electronic resource]. – Access mode: https://developer.nvidia.com/embedded/dlc/jetson-nano-developer-kit-user-guide (13.06.2020).

[6] CUDA C Programming Guide [Electronic resource]. – Access mode: http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (13.06.2020).

[7] N S Bakhvalov 1999 Efficient methods for stiff multidimensional multiparametric problems Comput. Math. Math. Phys. vol 39:12 pp 1938–1966

[8] A V Boreskov and E V Shikin 2014 Computer Graphics: from pixels to programmable graphics hardware (6000 BROKEN SOUND PARKWAY NW,STE 300, BOCA RATON, USA, FL, 33487-2742) p 543

[9] A V Boreskov 2019 Computer graphics programming. Modern OpenGL (Moscow: DMK Press Moscow) p 372

[10] Yu A Kropotov A A Kolpakov 2019 Research and Development of the Algorithm of Mixing Audio Data Streams for Heterogeneous Computer Systems in Telecommunications 2018 Dynamics of Systems, Mechanisms and Machines (Dynamics) DOI: 10.1109/Dynamics.2018.8601424

[11] A A Belov Y A Kropotov and A Y Proskurjakov 2010The analysis and processing of experimental time series in systems of automated control KpbiMuKo 2010 CriMiCo – 2010 20th International Crimean Conference Microwave and Telecommunication Technology, Conference Proceedings pp. 520-521

[12] V V Jacob C D R Ferreira and R Ferreira 2012 GPU optimization techniques applied to scale free gene regulatory networks based on threshold function Proceedings - 13th Symposium on Computing Systems, WSCAD-SSC 2012; Petropolis, RJ; Brazil 6391764 pp 57-64

[13] L Oteski G Colin De Verdiere S Contassot-Vivier S Vialle and J Ryan 2018 Towards a Unified CPU-GPU Code Hybridization: A GPU Based Optimization Strategy Efficient on Other Modern Architectures Advances in Parallel Computing vol 32 pp 390-399

[14] Y Yuttakonkit S Takamaeda-Yamazaki and Y Nakashima 2016 Performance optimization of light-field applications on GPU IEICE Transactions on Information and Systems vol E99D issue 12 pp 3072-3081

[15] K A Nekrasov S I Potashnikov A S Boyarchenkov and A Ya Kupryazhkin 2016 Parallel computing for general purpose graphics processors: text book (Ministry of Education and Science of the Russian Federation, Ural Federal University. Ekaterinburg: Publishing house of the Ural University) p 104

[16] Y A Kropotov A Yu. Proskuryakov and A A Belov 2018 A method for predicting changes in time series parameters in digital information management systems Computer optics vol 42 no 6 pp 1093–1100