UDC 681.2

THE CONFIGURATIONS, SOFTWARE AND HARDWARE OF THE PLATFORM
OMAP L-138 (CPU/DSP) FOR IMAGE ACQUISITION

Wawrzyniuk B., Sutkowski M.

Institute of Microelectronics and Optoelectronics, Warsaw University of Technology
Warsaw, Poland

Primary configuration of the board can be done by RS-232. It is needed to prepare hardware for booting of the operating system. This can be done by two modules loaded to the OMAP's memory with use of RS-232 port (main bootloader and U-Boot – universal bootloader). The first one initializes the hardware, sets-up clocks rates and loads the U-Boot. It performs necessary hardware tests, configures main parameters and initializes operating system loading from SD card. Manufacturer designates a Linux-based compilation to work with the platform. It has to be equipped with additional software to alow programming: D aVinci PSP 03.20.00.13 and complete Linux SDK for OMAP-L138 / TMS320C6748 EVM. Components necessary to include to the operating system are listed in Table 1.

With the above-described software, it is possible to write and run software on the OMAP-L138 platform. With use of C6RUN it is enabled to compile of codes written in the C language directly onto the DSP processor, or to create libraries from specific files. That makes it p ossible to compile the code with a cross-compilator in such a way, that the main function is launched on the GPP processor, and only its particular functions on the DSP core.

Below are presentest relusts of tests of sample operations run on different units, ARM, DSP or both ARM+DSP. Two type of operations were applied in the C code: mathematical calculation and working with file objects.

Calculation of PI number with use of Leibnitz formulae (function pilicz):

$$\sum_{n=0} \left(\frac{-1^n}{2n+1} \right) = \frac{\Pi}{4}$$

where: n – denotes the number of cycles. During the tests the number of cycles were $n = 500\,000$.

Operation on file objects (function plik). It is simple open, write a 20-element string and close text file. The number of cycles were $n_{file} = 500$.

Both operations were compiled in a w ay where all code was placed in single function main() – denoted as "pilicz()" and "plik()", and where inside main() function there was a c all for function placed in the external file – denoted as "pilicz_outer()" and "plik_outer()". The results shown as a duration of the execution of specific tests are shown in Tab. 2. As can be seen the skillfull implementation of the tasks to be executed on heterogenic processors allow to optimize the operation time. In the tests function "pilicz" was ended approx. 3x faster on DSP core and function "plik" was ended approx. 30x faster on ARM core.

Image acquisition and processing with OMAP L-138. In general the described unit allow to work with digital data and signals. Including image acqusition and processing. To demonstrate the possibilities of this platform a popular network camera Axis M1031 were used as image acqusition device. Due to the necessity of connection to the host computer the camera was connected to the board with use of external router (see Figure 1).

Table 1 – The list of the components of the system

DSPLINK	**It is software that provides communication between the DSP and ARM cores.**
CMEM	**This is the so-called application and library programming interface (APIs) used for managing memory blocks.**
CODECS	**Algorithms used for processing image and audio data on a signal processor.**
C6ACCEL	**It allows writing programs that process with the support of the DSP core**
DMAI	**Davinci Multimedia Application Interface**
C6RUN	**It is a project based on an open source license used to quickly launch (start?) applications on heterogeneous systems.**

Table 2 – Results of tests performed by OMAP L-138 cores.

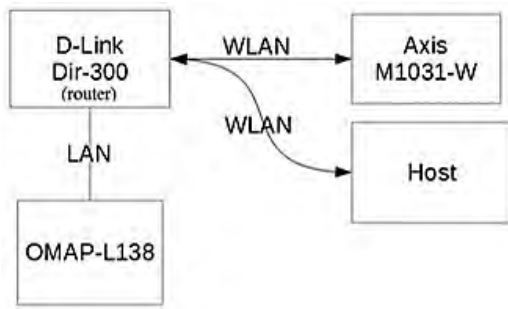| Core \ Function	Pilicz()	Plicz_outer()	Plik()	Plik_outer()
ARM	**5312 ms.**	**5302 ms.**	**269 ms.**	**270 ms.**
DSP	**1820 ms.**	**1981 ms.**	**8 s.**	**8 s.**
ARM+DSP	**5314 ms.**	**2243 ms.**	**271 ms.**	**8377 ms.**

Figure 1 – Network connecting diagram

Certainly the ARM/DSP board is able to work as standalone station if will be equipped with proper software. Below are some examples explaing how to work with image data using OMAP L-138 prototype platform.

The simplest way is to use a Gstreamer framework based on C [4]. It is a multimedia software working as a series of plug-ins. For acquire a video stream from the network camera a simple pipeline can be used:

gst-launch --gst-debug-level = 2 s rtspsrc location = 'rtsp: // <IP_CAM > /axis-media/media.amp 'latency = 100! queue! rtph264depay! filesink location = / home / root / video / test.264.

This command is capturing video data ovet Ethernet and saves on storage media as a common H.264-type video file. On Figure 2 there are a sample of the file captured by OMAP L-138 and played by host computer.
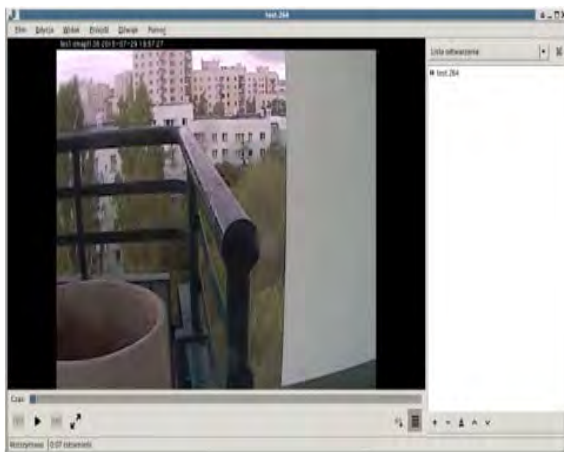

Figure 2 – Video stream captured and saved by OMAP L-138 prototype platform

Captured data can be also processed by OMAP L-138. Homogenous structure of the prototype platform is very suitable for this application. Using a proper design of the code with correct allocation of the tasks a real-time or quasi-real-time video processing can be peroformed. For example a DSP core can be effectively used for video format conver-

sions. The „TIViddec2" plug-in dedicated for DSP core is able to convert popular input formats (i.e. H.264, DivX, XviD, MPEG-2 etc.) to UYVY standard. A pipeline command converting stream to UYVY and saving to storage media can be found below and a resulting file is shown on Figure 3.
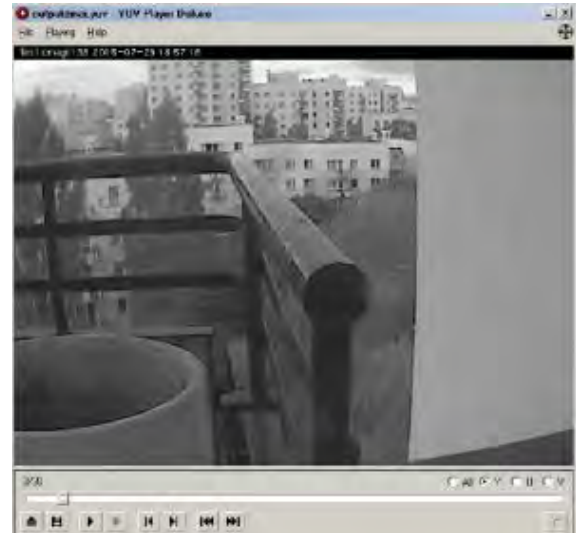

Figure 3 – Reconstruction of the luminance channel of the UYVY file

gst-launch --gst-debug-level=2 -e rtspsrc location='rtsp://192.168.2.105/axismedia/media.amp' latency=100 ! queue ! rtph264depay !queue !TIViddec2! filesink location=/home/ root/nagrania/test.yuv

With use of the specialized cores and appropriates methods of software design it can be created a very effective computational device for many different applications. Thanks to the low power consuption it can work even on battery and is lightweight and low heating. All the advantages makes similar architectures for portability.

The homogenous platforms can be effectively applied for the image capturing and processing. It is able to create together with capturing device (i.e. network camera) a standalone unit with image processing and analysis possibilities working even at real-time regime.

1. Hang K. Real-time Image Acquisition and Processing System Design based on DSP // ICCAE 2010. – Singapore, IEEE Press. – 2010.
2. Rongchun S., Yan P. Platform of image acquisition and processing based on DSP and FPGA // Appl. Mech. and Mat. – 2013. – V. 457-458.
3. Suran K. FPGA + DSP based real-time image acquisition system research and design // Adv. Mat. Research. – 2012. – V. 433-440.
4. link:http://gstreamer.freedesktop.org/documentation/ (access: 2015-08).