

Embedded Video Processing Based on Davinci Platform System Design

Gang Wang

Institute of Intelligent Vision and Image Information, China
Three Gorges University
YiChang, China, 443002
Sonny2005@qq.com

Jun Cheng

College of Computer and Information Technology, China
Three Gorges University
YiChang, China, 443002

Abstract-This paper main adopts TI company's TMS320DM6446 DAVINCI series chip designed for a set of video processing system. The system through the simulation of incoming video on A/D conversion, and then through the DSP end compression processing, thus realize real-time transmission and the large capacity of storage. Its hardware design part mainly include: The power modules, video processing module, network transmission module, etc. In this system can realize intelligent algorithm above the transplant, but also as a video processing system operate independently. Software is mainly ubl, uboot transplantation, the operating system's transplantation, codec engine.

Keywords-Davinci, Video processing, TMS320DM6446, Algorithm transplantation

I. INTRODUCTION

As people security requirements of growth and to monitor sites increases gradually, ordinary video monitoring the artificially already can not meet practical needs. Develop a good real-time, high stability, low cost, talgorithm portable video monitoring system is an inevitable trend. The paper presents the design of embedded video processing system have more requirements. TMS320DM6446 is the core processing chip, which is the speed reaching 4800 mips dual processor, ARM+DSP. ARM can very good of whole system control. DSP processors complete motion detection, video compression and target detection. Video processing chip is TVP5158, it has the adoption of image high-definition, and can be collected four images. Peripheral circuit design include: RS232 serial port, the SD card, phy, interface circuit. In addition, under the support of the relevant function library in TI, such as, video coding h.264, g711 speech coding can complete the video processing system.

II. THE INTRODUCTION TO TMS320DM6446

TMS320DM6446 is one of TI company DAVINCI series of dual-core processor chip, ARM processor is ARM926EJ-s core, working frequency is 297MHZ, DSP processors use is c64X +, working frequency is 597MHZ, speed reached 4800MIPS. The internal structure of the chip is a ARM subsystems, a digital signal processor (DSP) subsystem plus a VPSS, and some peripherals. The ARM is mainly responsible for the DSP subsystem, VPSS subsystems, and most peripherals and memory configuration and control. And DSP is mainly used for data processing, such as video coding and decoding. DSP eight function units composition, a total of 32 registers, they combined

together complete all the logic operation and the results will be loaded into the operation results of the register and deposited again storage medium. Its core structure is show as follows:

III. VIDEO PROCESSNG SYSTEM HARDWARE MODULE DESIGN

This video processing system is a video processing smallest system, we can put it into by the following structure. first, video processing front-end. second, network transmission module design. Third, video processing back-end design. Video processing front-end mainly is simulated by the video cameras collected transmission to a system of joint BNC decoding chips, then put the decoded data transmission to the system input port. Network transmission is DM6446 internal integration EMAC (Ethernet media access controller), through Ethernet physical transmitter , can conveniently network transmission, system upgrades, network access.Video processing back-end mainly has video encoder, the related calculation of data DSP and image display module (OSD). The whole system operation process can show: The input analog video signal through TVP5158 chips for A/D conversion and sent directly to VPFE (video processing front-end), a 256M 32-bit DDR DM6446 connected with the DDR, DDR starting .address set to 0X80000000. DDR memory controller through proprietary bus connected to the central exchange network (SCR), this can meet all the sons DM6446 module of DDR visit. In addition DM6446 through external hard drives can be compressed and video store.It shows as figure 2:

IV. VIDEO PROCESSING SYSTEM SOFTWARE ARCHITECTURE

Video processing system software parts; We will divided it into the following parts: first, The development environment. second, UBL, UBOOT transplantation. third. The operating system transplantation. Fourth ,codec engine environment, Dual-core communication and video decoding examples realization.

A. The building of development environment

The building of TI DAVINCI DM6446 development environment unlike Samsung S3C2410 ATMEL S3C2440 simple, because the development environment DM6446 include DSP and dual-core communication problems therefore the development strategy first step is to build good development environment. you should do as follows:

(1) Ready to all sorts of relevant Windows, Linux, DAVINCI software installed software setups; VMware-workstation, RedHat Enterprise Linux Server 5, mvl_5_0_0801921_demo_sys_setuplinux.bin, mvl_5_0_0_demo_lsp_setuplinux_02_00_00_140.bin, dvsdk_setuplinux_2_00_00_22.bin, bios_setuplinux_5_33_06.bin, xdctools_se_tuplinux_3_10_03.bin, ti_cgt_c6000_6.0.23_setup_linux_x86.bin

(2) DSP development tools, CCS3.3, tds560plus emulator, bios_setupwin32_5_33_06.exe

(3) Linux environment build; ARM compile environment build; DSP compile environment build.

B. The transplantation of ubl ,uboot

RBL (arm rom bootloader) burned in the ROM already, Because RBL only support of NAND FLASH 14k boot program and the compiled bin file commonly over 80K, especially version is high, at this time, so it need to write a UBL. UBOOT NAND FLASH read from UBL, put UBOOT COPY DDR2(RAM) to the relevant addresses, then boot the uboot. UBOOT transplantation is the necessary step, Here the version is uboot-1.3.4, we should delete our own development platform not related files and folders and link cross-compiling environment. Amendment to the top of MAKEFILE files. and transplant board driver and related configuration.

C. The operating system transplantation

The version of the operating system is montavista linux-2.6.18. First, what should do is delete some platform not related to platform, enter arch list and delete all other list, save arm list. Modify the arch/arm/on/makefile. Second, build cross-compiling environment, we should do is Kernel transplant cuts, using makemenuconfig command configure the kernel and remove the driver not related with platform, modify the arch/arm/Mach - davinc/board evm. Third step is saving backups modified configuration.

D. Codec engine environment

The Codec Engine is a software platform for algorithm execution, it can communicate between arm core and dsp core.

In TI's company's words, the Codec Engine is a set of APIs that you use to instantiate and run xDAIS algorithms. The API is the same for all of the following situations: The algorithm may run locally (on the GPP) or remotely (on the DSP), The system may be a GPP+DSP, DSP-only, or GPP-only system. All supported GPPs and DSPs have the same API. All supported operating systems have the same API. For example, Linux, PrOS, VxWorks, DSP/BIOS, and WinCE.

Codec Engine is a designed to solve some common problems, associated with developing system-on-a-chip (SoC) applications, The most significant problems include: Debugging in a heterogeneous processor environment, Changing to a more efficient algorithm involves significant recoding. Some algorithms may run on either the GPP or the

DSP. To balance system load, "low complexity" algorithms can run on a GPP, but the definition of "low" changes over time. If changing the location where the algorithm runs were easy, you wouldn't have to weigh performance issues against the difficulty of changing the application.

In our platform, we also need using codec engine. For this video analysis system, we first need to DSP engineers use CCS develop their own video decoding algorithm, compiler generates a decoding algorithm library file *.lib. and then generate a generation to run on another DSP executable *. X64p (namely. Out files) is also DSP Server. The third step, based on DSP Server name and contains specific video decoding algorithm create Codec configuration file *. cfg, Finally, application engineer received different CODEC pack, DSP Server and started the configuration file *. Cfg, application through compiler, links, ultimately make ARM side executable file.

V. EXAMPLES ON THE REALIZATION OF EVALUATION VERSION

First, we should have our DVDEM properly set up, and we are able to mount an NFS, also we should pass the MEM=120M parameter to your linux kernel from your prompt. In an NFS directory visible from the DVEVM board, copy all the files found in apps/system_files/<platform>. Those files are dsplinkk.ko, cmemmk.ko, loadmodules.sh. Copy to the same directory as above the pre-built DSP server and GPP client executables. also copy the sample video file to the same directory, app/sanity_test/<platform>/video_copy.x64p, app/sanity_test/<platform>/app.out, app/sanity_test/<platform>/in.dat.

Second, boot the EVM, change to the directory where you have copied all the files, and run sh./loadmodules.sh, next, run the client application, which will automatically load the DSP server image: ./app.out, you will see as follows[2]:

VI. SUMMARY

This paper adopts TMS320DM6446 finished video processing system design. It can realize video D1 standard acquisition, storage, and real-time compression. Its performance is superior than traditional monitoring which is pc add acquisition card. low consumption, more stable, easy to large storage video data, and low cost. The video processing system based on DM6446, further can also transplant intelligent monitoring algorithm. For example, license plate detection, face recognition, etc.

ACKNOWLEDGMENT

This project is supported by National Natural Science Foundation of China (60875009, 60972162), Major Program of Educational Commission of Hubei Province of

China(Z20081301) and the Natural Science Foundation of Hubei Province (2008CDB346), Program of science and technology R&D project of Yichang, China (A09302-31, A09302-32,A2011-302-17).

REFERENCES

- [1] Texas Instruments Incorporated. TMS320DM6446 Digital Media System - on - Chip [EB/OL]. <http://focus.ti.com.cn/docs/prod/=folders/print/tms320dm6446.html>, 2007- 03- 05.
- [2] Texas Instruments Incorporated.Build /Run Instructions For Codec Engine Examples Last updated April21,2009
- [3] Texas Instruments Incorporated.DVEVM TMS320DM6446 DVEVM v2.0 Getting Started Guide,Literature Number:SPRUE66 March 2006

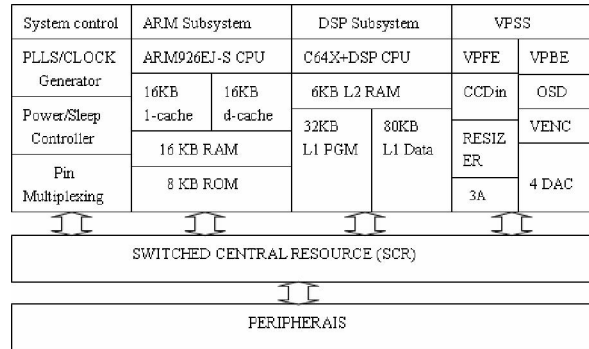


Figure1. Internal structure of system on chip about DM6446

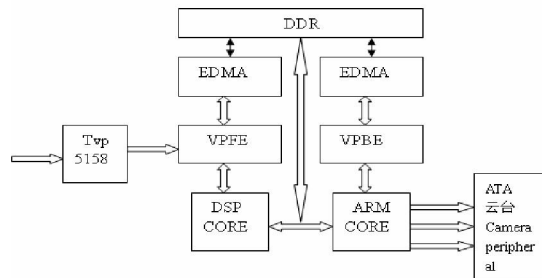


Figure 2. video data flow

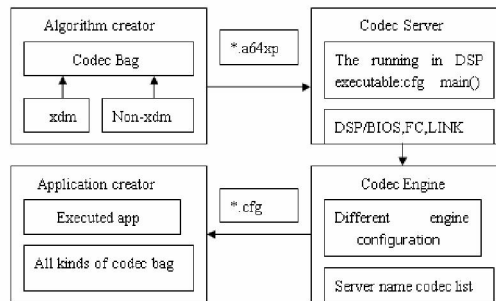


Figure 3. algorithm running on dm6446 need four steps

```

App-> Application started.
CEapp-> Allocating contiguous buffer for 'input data' of size 1024...
CEapp-> Contiguous buffer allocated OK (phys. addr=0x87fff000)
CEapp-> Allocating contiguous buffer for 'encoded data' of size 1024...
CEapp-> Contiguous buffer allocated OK (phys. addr=0x87ffe000)
CEapp-> Allocating contiguous buffer for 'output data' of size 1024...
CEapp-> Contiguous buffer allocated OK (phys. addr=0x87ffd000)
App-> Processing frame 0...
App-> Processing frame 1...
App-> Processing frame 2...
App-> Processing frame 3...
App-> Processing frame 4...
App-> Finished encoding and decoding 4 frames
App-> Application finished successfully.
    
```

Figure 4. algorithm successfully running on our own platform