Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 812356, 10 pages
doi:10.1155/2010/812356

Research Article

Power-Aware DVB-H Mobile TV System on Heterogeneous Multicore Platform

Yu-Sheng Lu,[1,2] Chin-Feng Lai,[1] Chia-Cheng Hu,[3] Han-Chieh Chao,[4] and Yueh-Min Huang[1]

[1] *Department of Engineering Science, National Cheng Kung University, No.1, University Rd., Tainan 701, Taiwan*
[2] *Business Customer Solutions Laboratory, Chunghwa Telecom Laboratories, No. 12, Lane 551, Min-Tsu Rd. Sec.5 Yang-Mei, Taoyuan 326, Taiwan*
[3] *Department of Information Management, Naval Academy, No. 669, Junxiao Rd., Zuoying District, Kaohsiung 813, Taiwan*
[4] *College of Electrical Engineering & Computer Science, National ILan University, No. 1, Sec. 1, Shen-Lung Rd., I-Lan 260, Taiwan*

Correspondence should be addressed to Yueh-Min Huang, huang@mail.ncku.edu.tw

Received 19 March 2010; Accepted 15 June 2010

Academic Editor: Liang Zhou

In mobile communication network, the mobile device integrated with TV player is a novel technology that provides TV program services to end users. As TV program is a real-time video service, it has greater technical difficulties to overcome than a traditional video file download or online streaming, especially when TV programs are played on handheld devices. A challenge is how to save power in order to provide users with longer TV program services. To address this issue, this study proposes a mobile TV system on a heterogeneous multicore platform, which utilizes a Digital Video Broadcasting-Handheld (DVB-H) wireless network to receive the TV program signal, thus, saving power according to the features of DVB-H TV signal and heterogeneous multi-core.

1. Introduction

Along with the progressive digital TV broadcasting technology, TV viewing is no longer restricted by time or space; the new trend is to watch digital TV programs through wireless mobile devices. At present, watching TV on a mobile phone device can be performed in two ways. Service providers can transmit TV program data to mobile phone users by 3G network, or a base station can transmit TV programs through a Digital Video Broadcasting network [1–3]. The main difference between 3G and DVB network is that 3G network transmits data through on demand wireless network communication, hence, there will be transmission rates and bandwidth limits if too many users access this network at the same time. As to DVB-H, it transmits TV programs through the broadcasting transmission of TV base station; hence, there will be no transmission network congestion. Three issues have been studied regarding the mobile TV system: (1) TV signal transmission technology and how to enhance TV signal fault-tolerance or increase signal transport efficiency in order to improve display quality of TV programs; (2) mobile TV application developments and provision of personal context aware services, recommending suitable TV programs according to user habits and preferences of watching TV; (3) how to enhance display quality, provide smooth TV programming if delays occur, and reduce power consumption in mobile TV players [4–7]. Concerning power-saving issues, two parts are discussed: (1) components of receiving TV signals, how to design receiver startup schedule while receiving a TV program signal to save receiver power; (2) design a power-saving play mechanism according to TV program signal features, after received TV signal is converted to digital data by the demodulator (Figure 1) [8–11]. Therefore, this study proposes a power-aware DVB-H mobile TV system on a heterogeneous multicore platform. This system is implemented in two major parts: a front-end buffer control mechanism and a parallel DVB-H TV signal decoding model.

When receiving a DVB-H TV program signal from a base station, signal is demodulated to generate video and audio data. As video bit rate, quality, and resolution are directly related to content complexity, running too many buffers will consume power, while too few buffers will cause the program to fail to be played successfully. Hence, this

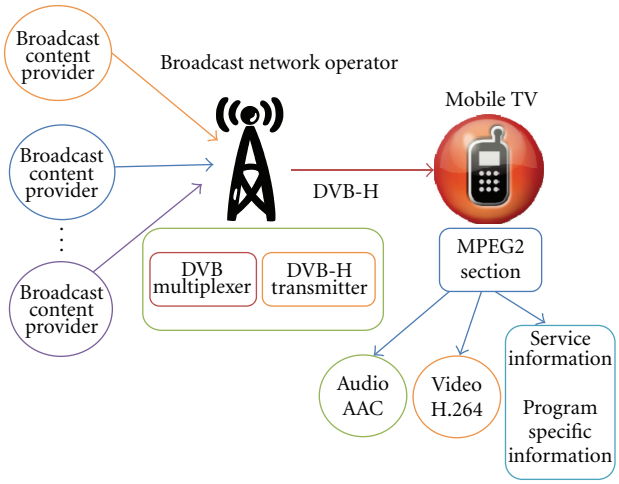

Figure 1: DVB-H Mobile TV Workflow.

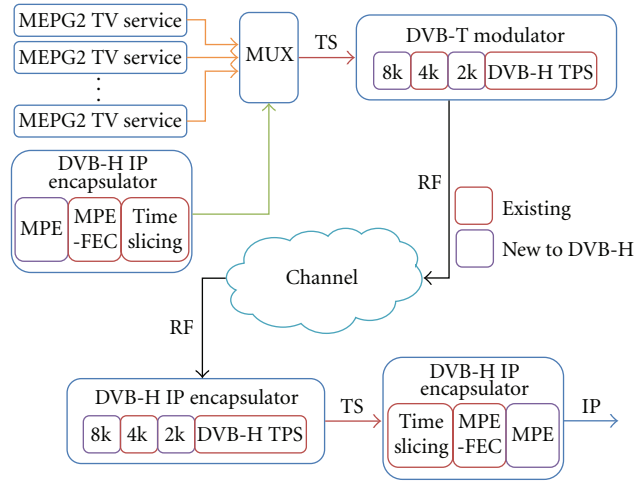

Figure 2: DVB-H/T System Archietcture.

paper proposes a front-end buffer control mechanism to configure appropriate buffers according to the TV program video features, in order to utilize buffers and save power.

The parallel DVB-H TV signal decoding model uses a data partition processing method to run parallel DSP decoding of DVB-H videos on a heterogeneous multicore platform. It also schedules videos according to the DVB-H video features, in order to reduce data dependency among the frames on a multicore platform.

The remainder of this paper is organized as follows. Section 2 introduces DVB-H specification and the parallel decoding technique; Section 3 presents the overall architecture of the DVB-H mobile TV system, the front-end buffer control mechanism, and the processes and methodology of the parallel DVB-H TV signal decoding model; Section 4 discusses implementation and result, and Section 5 gives conclusions.

2. Related Work

2.1. Digital Video Broadcasting-Handheld. Digital Video Broadcasting-Handheld (DVB-H) is based on Digital Video Broadcasting-Terrestrial (DVB-T) specification and provides a solution to lower receiver power consumption and improves mobile receiving performance [12–18]. Figure 2 shows the outline of the DVB-H/T system specifications for common TV broadcasting programs using the DVB-T signal transfer mode. Senders can use an A/D converter to convert the analog video and audio signals to a digital signal, respectively, and use a Moving Picture Experts Group 2 (MPEG-2) codec technique to convert TV program data into MPEG-2 format. DVB-H service data are compressed and encapsulated into an IP packet then encapsulated into the transmission stream through a Multiprotocol Encapsulation (MPE) mechanism. Meanwhile, the time slicing data stream is added. Along with other DVB-T TV services, the multiplexer multiplexes it into a larger transmission stream (or multiple program transmission stream) before sending the data in a DVB wireless network. At the receiver, if a client wants to receive certain services, the receiver front-end circuit must run continuously in order to obtain the complete transmission stream. Then, the demultiplexer extracts the video, audio, and data information streams of the selected programs and delivers this information to the video decoder, audio decoder, and other applications for processing. The sender Multi-Protocol Encapsulation-Forward Error Correction (MPE-FEC) and time slicing mechanisms are collectively called the DVB-H IP-Encapsulator, while the receiver reverse recovery portion is called the DVB-H IP-Decapsulator. The overall DVB-H container format is shown in Figure 3. The IP data container format for each layer of DVB-H is shown in Figure 3, as an IP packet in the MPE section and redundant data in the FEC section. After Section format encapsulation, the MPE and FEC sections are connected end to end according to the encapsulating sequence to form a section data string. Then, it begins to slice the first and all of the other 184 bytes of each section data string. A 4-byte transmission stream header is added to the front of the 184-byte data length in order to complete a transmission stream encapsulation or MPEG2 transmission stream packet. Its data length is 188 bytes, with two major parts. The first is a data front-end header that occupies a 4-byte length with the available information, including a Sync. Byte = 47 hex for synchronizing the emitter and receiver, error indications, and stream packet recognition. The second part is the data transfer payload, which length is 184 bytes. In Figure 3, above the IP packet is the User Datagram Protocol (UDP) and the Real-time Transport Protocol (RTP). The top layer is compressed video data, where IP and UDP packets add their packet headers. The RTP packet is encapsulated and used to bear the H.264 images and AAC compressed voice, as RFC3984 specification.

2.2. Parallel Decoding. One ideal parallel process could double the system processing efficiency; however, when coding/decoding a picture, there exists a data dependency problem [19–25]. As the video image format contained in the DVB-H TV signal is an H.264 baseline format, this section

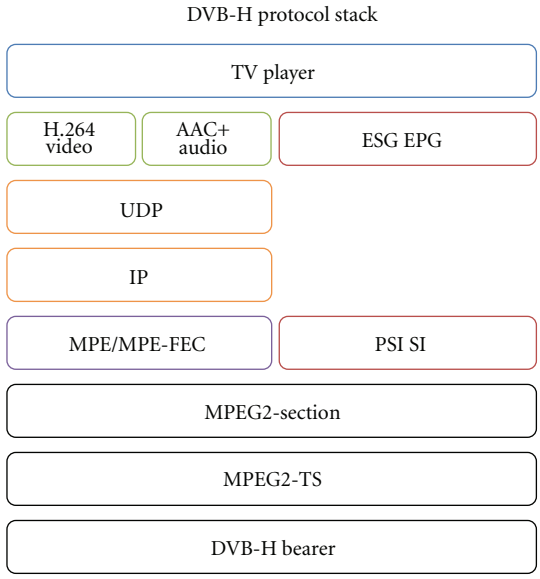

Figure 3: DVB-H Protocol.

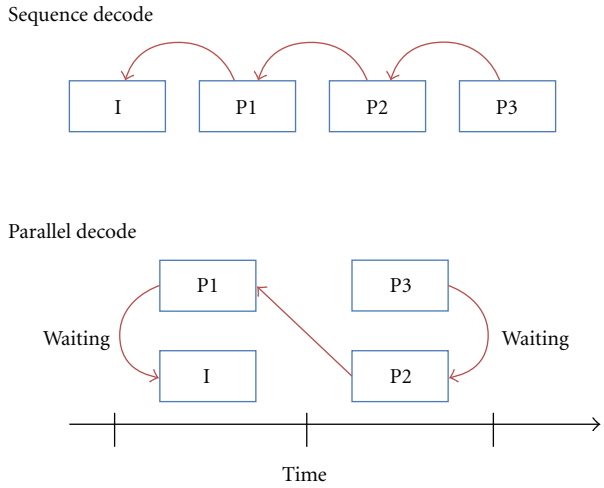

Figure 4: Sequence/Parallel Video Decode.

introduces the H.264 image feature. In H.264 decoding, pictures are divided into I frame, P frame, and B frame, where P frame is decoded according to the I frame picture data, and the B frame refers to the picture data of the I frame and the P frame. Unless there is good parallel processing, data collision will occur, as shown in Figure 4. When decoding two interdependent pictures, even when both pictures are simultaneously processed, the other picture must wait for a decoded reference before decoding. Therefore, how to utilize parallel processing to shorten the operation waiting time is the focus of many studies. Parallel decoding is divided into two orientations, a function partition, and a data partition, detailed as follows.

2.2.1. Function Partition. The H.264 decoding process can be roughly divided into entropy decoding (ED), inverse

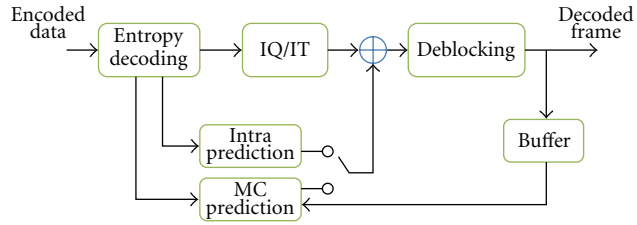

Figure 5: Functional Partition Decode Workflow.

quantization and inverse transform (IQ/IT), intra- or inter-prediction (PPC), and deblocking filter (DF). As shown in Figure 5, the function partition divides the entire H.264 decoding process into independent tasks. The main purpose is to use a balanced processing concept to configure the tasks required by each processor so that each processor can share the load, and processing can be accelerated. The advantage of this mechanism is that it can easily and extensively eliminate data dependency, while its disadvantage is that its task division is subject to a number of processors.

2.2.2. Data Partition. Data partitioning divides decoding data into partitions, which are computed by different processors. Each processor performs the same data operations, but process different data units. Previous literatures have studied how to divide data while avoiding data dependency; the partition includes groups of picture (GOP) levels, frame levels, slice levels, and macroblock levels. Their features are detailed below.

GOP level [21, 22]: it divides the video segments in GOP, allocates each GOP to each processor to decode, as each section of the GOP can independently run decoding. Decoding in this manner can increase processing quantities at linear speed [23]; however, applying this technique requires large memory space to save the decoded GOP fragments.

Frame level: this parallel decoding method allocates each single picture to a respective processor to operate, where preanalysis sorting operations or a tournament algorithm is adopted in order to process picture allocation. Primarily, two pictures without data dependency are found and simultaneously operated. Flierl and Girod [24] proposed a B Frame parallel decoding method, which is suitable for traditional coding methods, and because the B frame is not referred to by other picture, no data dependency will occur. B frame is analyzed first and allocated to different processors for decoding. However, regarding H.264 coding, B frame can be a reference for other pictures, therefore, is not suitable here.

Slice level [25, 26]: in H.264, the Slice is the smallest independent decoding unit, meaning that a single Slice can independently run decoding. Therefore, similar to GOP level partitioning, various Slices are allocated to various processors for decoding. As compared with the GOP Level, this parallel method is more favorable to memory utilization without extra analysis sorting. However, its main disadvantage is that, slice divisions can range from a macroblock to one entire frame, where memory use, scalability, and balance