System Architecture for a Multi-Media enabled Mobile Terminal

Stefan Drude, *Member*, IEEE, Manfred Atorf, Laurent Chivallier, Kenneth Currie

Abstract — *A system architecture for mobile terminals is proposed, in which the signal processing for the cellular communications channel and the associated protocol stack processing are implemented independent of the application system, which handles audio and video processing functions and the user interface. This approach enables the use of open operating systems for the application system and the integration of connectivity communication pipes into the system, while maintaining the integrity and huge investment in the protocol stack software of the cellular subsystem. High speed digital serial links are used to connect the processing subsystems and the video peripherals. This improves the modularity of the solution and gives the equipment manufacturers increased flexibility*[1].

Index Terms — Mobile terminals, cellular systems, wireless connectivity, digital serial interconnect, audio video streaming .

I. INTRODUCTION

Cellular phones and mobile terminals for cellular services have become fairly popular. In many countries the penetration rates for these devices have reached very high levels. The market for mobile phones continues to grow and is expected to reach more than 2500 million subscribers worldwide by 2008. According to our estimates, about 650 million mobile phones will have been sold in 2004, a 25 percent increase over 2003.

A number of different cellular communication standards are being used across the world. The most common standards today are all based on digital communication techniques. Despite their specific differences, the basic architecture of a voice centric phone is always the same. The main functional blocks include the analogue subsystem with its radio transceiver, the baseband signal processing along with the protocol stack processing function, and finally the user interface.

The cellular network operators are increasingly interested in offering services beyond the more traditional voice centric services. This will allow them to differentiate their cellular products and services against their competitors and to maintain a reasonable average revenue per user. As a result, new features enter the mobile terminal, which are needed to enable and support these new services. For example, camera and image sensors are needed for Multi-Media Messaging Services (MMS), high quality audio player capabilities are required to play downloaded music files. Future mobile terminals will even include receivers for digital television signals. These new features usually also require larger color displays with higher resolution, higher frame rate, and more supported colors. In addition, consumers expect a change from cables and wired connectivity for the peripheral devices around their mobile terminals to wireless connectivity. Also, network operators are seeking to offer alternative ways to connect to their networks in dense environments through the installation of wireless hotspots based on wireless local area network (WLAN) technology.

The increasing demand for multimedia capabilities and support for demanding high end applications in future mobile cellular phones has made a change in the chipset architecture necessary. The processing of audio records, still images and even live video streams, while running complex application software at the same time, is difficult to support with a cellular baseband device, which has been designed primarily to handle cellular signals and the associated protocol stack processing.

In this paper we present our proposal for a dual processor architecture, which is discussed in section II. The central media processor, the "application engine", and its main processing blocks are reviewed in section III. The media processor software is based on an open operating system; the software architecture for such a distributed processor system is explained in more detail in section IV. We have based our system architecture on a network of fast digital serial links. The requirements and design choices for these links are presented in section V. Finally, section VI addresses the integration of the various broadcasting and networking pipes into the overall system architecture and outlines a path towards future convergence.

II. DUAL PROCESSOR ARCHITECTURE

The architecture of a voice centric, low end mobile phone does not usually offer an easy migration path towards a multi-media capable high end or smart phone. In many designs it is not uncommon to implement the protocol stack processing software based on a dedicated, low footprint operating system, which offers only a minimum set of real-time functionality to the software programmer. This makes it very difficult to add sophisticated audio and video streaming functionality while maintaining predictable real-time performance for both, protocol stack processing and media streaming applications. Also, in order to save cost, the processor used may not offer some of the hardware features needed to support more capable, standard operating systems for embedded systems. The

[1] S. Drude is with Philips Semiconductors, 5600 JZ Eindhoven, The Netherlands (e-mail: Stefan.Drude@philips.com)

M. Atorf is with Philips Semiconductors, 90443 Nürnberg, Germany (email: Manfred.Atorf@philips.com)

L. Chivallier is with Philips Semiconductors, Sophia-Antipolis, 06560 Valbonne, France (email: Laurent.Chivallier@philips.com)

K. Currie is with Philips Semiconductors, 6543 AE Nijmegen, The Netherlands (email: Kenneth.Currie@philips.com)

Manuscript received March 26, 2005

solution presented in this paper is based on a system architecture, which employs two separate processors to deal with protocol stack processing and media streaming, respectively (fig.1).

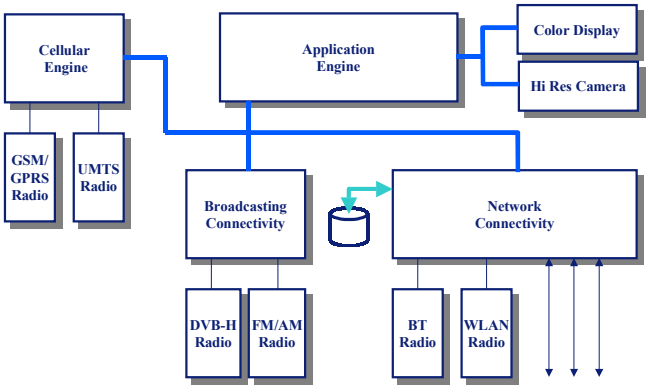

Fig. 1: High Speed digital links connect the various subsystems. Several different types of serial links will be employed in the proposed architecture to reflect the specific needs of the subsystems to be connected.

The first processor, the cellular engine, focuses on the protocol stack processing for the cellular system, it can be largely based on the software solution for a voice centric phone. This approach offers the extra benefit that the investment made in the protocol stack processing software and its type approval can be safeguarded.

The second processor in the proposed architecture, referred to as the application engine, is used to provide the multi-media related processing functions such as audio codecs, image processing and compression, video codecs, and the man-machine interface. It becomes the central media processing and master control unit in the mobile terminal. The application engine connects directly to all video peripherals and the cellular engine, and has also a link to the connectivity subsystems for networking, storage and broadcasting functions (fig. 1). This processor is intended to run standard, open operating systems in order to facilitate the implementation of time critical audio and video streaming functionality using software components available for these operating systems. The software structure will be further explained in section III.

The application engine also implements the switchboard of the system managing all the incoming and outgoing streams for the various applications. That way, the resources of the application engine can be equally used among the communication pipes connected to the mobile terminal and the applications using them (fig. 2). As can be seen from fig. 2, the application engine will typically also be used to implement the less time critical parts of the higher layer protocol stack processing for some of the communication pipes connected to it. As an example, the Bluetooth subsystem will normally implement the protocol stack processing up to the host control interface (HCI), while the application engine implements the higher layer protocol processing and particularly the various profiles to be supported by a specific product.

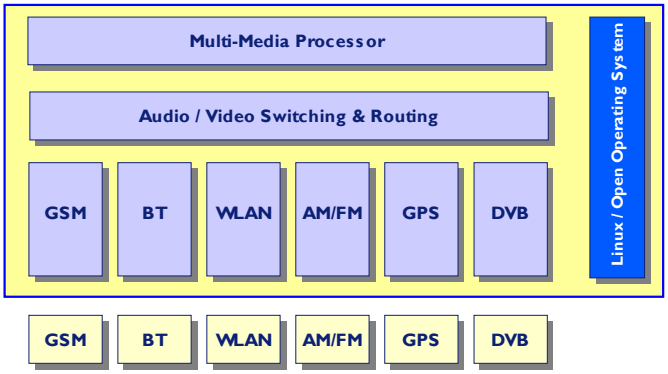

Fig. 2: High level media streaming architecture. The application engine is used to implement the audio routing and switching functionality in addition to being the main media processing unit.

The proposed architecture offers a high degree of flexibility to the mobile terminal manufacturers in that it allows to add and remove communication pipes supported by a particular implementation and product as needed. That way, a whole range of mobile terminals can be designed and built, which can cover various target markets with a single underlying hardware architecture. Also, with the separation of the cellular protocol stack processing from the media streaming and application software, the system designer can develop the two main subsystems largely independently, has more choice for operating systems and software components, and can develop a mobile terminal with good and predictable real-time performance. As an additional benefit, the rate of change can be made different for the cellular engine and the application engine, which helps to follow new market trends quickly.

III. APPLICATION ENGINE

The processing tasks associated with media streaming and man-machine interface functions can best be split into control specific functions and signal processing algorithm oriented functions. That way, the optimum hardware structure can be used for every function, which is a prerequisite to achieve low power consumption. The architecture of the application engine, which assumes the role of the master control unit in the mobile terminal, reflects this basic assumption. A general purpose RISC processor (Reduced Instruction Set Computer) is used to implement the control specific functions, a programmable digital signal processor is provided for audio specific signal processing. Dedicated hardware accelerator blocks are provided to perform specific video processing tasks (fig. 3). The internal communication is based on a multi-layer parallel bus structure, it employs a master slave structure and allows for multiple transactions to go on in parallel. Note that the partitioning outlined in this paper provides for independent future evolution of control, audio, and video functionality.

A. General Purpose Processor

The RISC processor subsystem consists of a number of blocks, which are carefully selected and tuned to achieve a

good balance between overall system performance and power consumption. The processor itself is an ARM926EJ processor with integrated memory management unit, which is needed to efficiently run more dedicated operating systems. Cache memories are provided for data and instructions so as to minimize access to the external memory. The subsystem further contains an interrupt controller, SRAM and boot ROM, timers, and provisions for testing such as an embedded trace buffer memory. The RISC processor is clocked at 208 MHz, which provides sufficient processing power to cope with the tasks at hand.

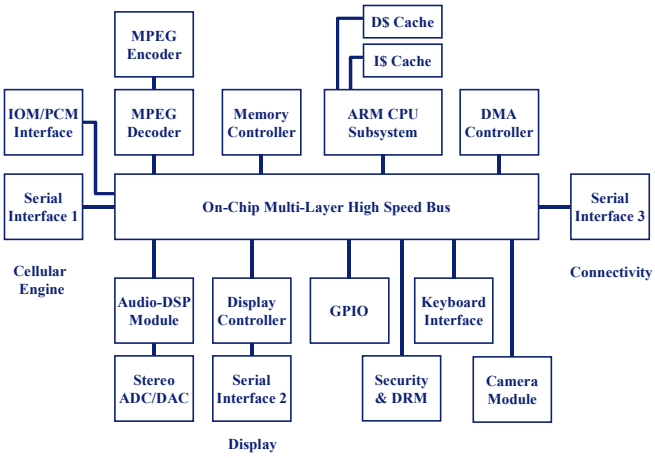

Fig. 3: The application engine features a number of dedicated subsystems to support the desired media processing tasks. The audio subsystem is based on a programmable digital signal processor (DSP) while hard-wired accelerator blocks support processing of MPEG coded video information.

A dedicated memory controller is provided to support a variety of memory types usually found in mobile terminals: NAND Flash memory for program storage, SDRAM (synchronous dynamic random access memory) for data storage, and in addition dedicated interfaces for removable flash memory cards such as MMC & SD Cards and memory sticks. The memory controller is supported by a direct memory access unit (DMA), which not only implements fast data transfers without using any processor time, but which provides also a special update mechanism for the liquid crystal display unit explained later.

B. B. Audio Processing Subsystem

The audio processing subsystem is based on a 16-bit digital signal processor and is accompanied by a set of dedicated audio converters to handle high quality stereo input and output. The DSP features on chip program and data memory, the RISC processor will load the program memory before execution of DSP instructions can start. Typical functions implemented on the DSP subsystem include audio codecs such as mp3, AC2 and AAC along with the various voice codecs used in voice over internet (VoIP) applications. In addition, the DSP subsystem will provide audio enhancement functions such as background noise reduction, echo cancellation, dynamic bass boost, polyphonic ring tones, and others.

C. Video and Graphics Subsystem

The specific video processing functions are distributed over multiple dedicated hardwired blocks. Video processing is targeted for MPEG-4 coded streams, two dedicated accelerator blocks implement the associated encoding and decoding functions. Compressed images are decoded by the JPEG decoder hardware, which can decode 1024 x 768 pixel images at a rate of eight pictures per second. The 3D graphics engine and display controller is needed to support rendering and texturing functions needed by state of the art gaming applications. The camera module deals with the camera interface itself and implements the pre-processing needed for still image and streaming video applications.

D. Power Management

The various clocks needed to operate the application engine are all derived from a central 26 MHz clock signal. Frequency multipliers provide clock signals of 208 MHz, 104 MHz, and 52 MHz, respectively. Whenever a particular block in the application engine is not needed to perform a particular task, the clock signal to this block is turned off to minimize power consumption. This approach has been implemented on several levels turning off macro blocks as well as smaller processing blocks within a macro block. The power management function is controlled by the general purpose processor through a dedicated power management control block.

IV. SOFTWARE STRUCTURE & ECOSYSTEM

The software structure proposed in the new architecture is based on a modular approach. The software is designed as a distributed system with multiple processors. In the present proposal, the cellular and connectivity communication pipe subsystems each handle at least part of the standard specific protocol stack processing independent of the application engine software. This architecture ensures that the investments made in legacy software for the cellular system protocol stack processing can be safeguarded and incorporated in the new system solution. It also allows for decoupling of the real-time critical tasks between the communication subsystems and the multi-media processing tasks, i.e. the signal processing tasks of the cellular subsystem do not compete with the audio/video processing tasks for compute power.

Standard application programmer's interfaces (APIs) are defined between the application engine and the respective communication pipe subsystems; they may even be defined as part of the standard itself. In some cases, the APIs are chosen such that the application engine will implement the upper layer protocol stack processing

A. Operating System Choice and Ecosystem

An open operating system is suggested for the RISC microprocessor of the application engine, which makes the integration of the various middleware and application software components easy compared to the integration with the legacy cellular protocol processor stack and its associated proprietary

operating system. The choice of open operating system is governed by the individual needs of the manufacturer and time to market requirements. Popular candidates for operating systems are Windows CE, Symbian and Linux, which is available through the open source community. One clear advantage of going for an open operating system over proprietary solutions is the availability of skilled resources and application software components in the open market. This gives the system integrator more flexibility in the use of software resources without the need for training, which is normally needed with proprietary solutions.

All three operating systems have a fairly large established ecosystem around them, which ensures that many software components are readily available from either a number of commercial suppliers or through the open source community. The system integrator can decide for each component whether to develop it in house or buy it, which helps to manage time to market requirements. This is a clear advantage compared to proprietary operating systems, where neither skilled resources nor off the shelf software components are readily available.

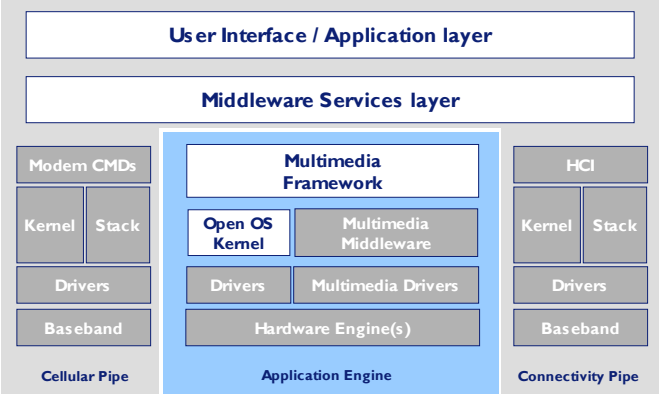

Fig. 4: The software in the proposed system architecture is distributed across the various subsystems. In this example, the cellular pipe and the Bluetooth subsystem are shown. The same approach is also used for the remaining connectivity subsystems.

B. Software Structure

The software structure for the dual processor architecture is shown in fig. 4. We will first focus on the software structure for the application engine. The open operating system on the application engine has been discussed in the previous section, dedicated drivers connect the operating system to the specific hardware blocks such as timers, memory controller and management unit, display controller, input/output devices, etc. The resulting platform layer will provide the basic, general purpose functionality needed to boot the system and to provide standard text based user interface functions.

The middleware layer combines both, the application and networking framework needed for mobile terminals. The application framework focuses on functions needed to implement the graphical user interface along with basic text and graphics handling primitives. As a subset of this framework, basic functions needed to handle the media related functions are also implemented such as encoding, decoding, and rendering functions for audio and video and still images. This block can be further enhanced by security and digital rights management functions, for which dedicated hardware accelerators are also included in the application engine discussed earlier.

The networking related framework provides the upper layer functionality needed to operate all the communication pipes connected to the application engine. At the lowest level, this includes the remaining parts of the protocol stack processing, for example the Bluetooth protocol stack processing and profile implementations from the HCI interface upwards, and the GSM control interface. Further up, we would find the standard communication protocols such as the Transfer Control Protocol (TCP), Internet Protocol (IP), Hypertext Transfer Protocol (HTTP), and others.

At the services layer we find the Java infrastructure and the application framework. We have chosen to base our application framework on Java and the Java 2 Platform Micro Edition (J2ME) virtual machine. This engine is specifically designed for small, embedded systems requiring a low footprint. The use of Java provides for a hardware and operating system independent solution, which makes porting of the application software to other hardware platforms and re-use from generation to generation easy. With its standardized software API, many software components are available for this platform, and there is a large software developer community working on this platform. As an additional benefit, software development can be started on a personal computer before the tested software is ported onto the target system.

The application framework implements all functionality needed to efficiently run and execute application software. This includes the various tools such as address books, calendars, business applications, but also the graphical user interface to the cellular phone and other communication functions. Most commercial operating systems provide enough means for customization at this level so as to make sure that a manufacturer can implement a brand specific look and feel for his products.

Finally, at the application level, the graphical user interface and the application frontend functionality are implemented.

C. Distributed Software Processing

The cellular and connectivity subsystems each implement their share of the baseband and protocol processing for a specific standard. From a software architecture perspective, these subsystems are remote units, which communicate with the application engine through dedicated interface channels. For the cellular subsystem, the interface is based on the TS–27.010 protocol. The cellular subsystem is loosely coupled to the application engine through this interface and executes the majority of the baseband and protocol stack processing independently. The TS–27.010 interface provides a set of high level modem type commands, which are used by the application engine to initiate and manage calls, send and

receive messages, manage the system level functionality of the cellular subsystem. Media streaming can go on via this interface in parallel to the cellular modem control functions.

The Bluetooth subsystem follows a similar approach, except that the interface follows the Host Control Interface (HCI) standard defined by the Bluetooth Special Interest Group. The HCI operates at a much lower layer in the protocol stack hierarchy, and consequently more of the Bluetooth protocol stack functionality and in particular the application specific profiles have to be implemented in the application engine software.

V. COMMUNICATIONS INFRASTRUCTURE

The various subsystems of the proposed architecture are connected through digital serial communication links; the bridging function between these links is implemented in the application engine. This approach offers a number of advantages over parallel interfaces in that it allows for more modularity and flexibility, serial connections greatly simplify clock data synchronization and can operate up to higher data rates, the amount of interference generated by on board communication is significantly reduced, and the board layout and interconnect wiring are greatly simplified. As will be further explained in the following sections, a mixture of dedicated, point to point links and serial bus oriented links will be used to reflect the real-time and communication needs of the various subsystems. The speed and clock rate of the serial links need to match the requirements of the data to be sent across the link. Not all interfaces have the same requirements. From there, it is foreseen that a number of different serial link types shall be implemented.

A. Display and Camera Requirements

Generally speaking, display and camera subsystems have the most stringent requirements on data rate and throughput. With the continuing increase in resolution and frame rate for displays and camera sensors, the data rates for the display subsystem have passed the 500 Mbps mark when accounting for some system overhead. A summary of display sizes and frame rates and the resulting data rates is shown in table 1. For the design of our system architecture, we assume that VGA resolution at 24 bit color depth and frame rate of 60 Hz in combination with mega-pixel camera sensors will become very common for smart phones and high end value added services terminals.

In addition to the requirements for high data rates, display and camera subsystem also require that the data be carried continuously and at low latency. Any delay in the delivery of the streaming display or camera information will result in artifacts on the screen, unless adequate buffering is provided in the devices, which is not desirable due to the associated cost and a likely increase in power consumption. It is for this very reason that dedicated links are proposed for the display and camera subsystems.

	H(V)	V(H)	color depth (bits)	frame/s	overhead	Mbps
QQVGA-	130	130	18	60	1.20	21.90
QQVGA	160	120	24	60	1.20	33.18
QCIF	176	144	24	60	1.20	43.79
QCIF+	176	208	24	60	1.20	63.26
QCIF+	176	220	24	60	1.20	66.91
QVGA	320	240	24	60	1.20	132.71
CIF	352	288	24	60	1.20	175.18
CIF+	352	416	24	60	1.20	253.03
CIF+	352	440	24	60	1.20	267.63
(1/2)VGA	320	480	24	60	1.20	265.42
(2/3)VGA	640	320	24	60	1.20	353.89
VGA	640	480	24	60	1.20	530.84
WVGA	800	480	24	60	1.20	663.55
SVGA	800	600	24	60	1.20	829.44
XVGA	1024	768	24	60	1.20	1358.95

Table 1: Display and camera parameters and resulting data rates. A typical VGA display will require more than 500 Mbps data rate, when playing a video stream at 60 frames per second.

A number of new technologies could also fall into this category, but are not covered in more detail in this paper. High data rate ultra wideband communications currently being standardized under IEEE 802.15.3a, future high speed extensions of the IEEE 802.11 standard, serial memory devices and dedicated additional co-processors are all candidates for particularly high speed links in order to achieve the desired throughput and to make the nature of the interface transparent to the system designer.

B. Communication Subsystems Requirements

The various wireless pipes connecting the mobile terminal to the outside world impose less stringent requirements on the serial interconnect link than display and camera devices. The underlying justification is based on the fact that, by the time the information is carried across a wireless pipe, it is usually compressed and efficiently encoded, and the wireless systems today offer limited bandwidth for communication. In addition, we need to account for system level overhead, which further reduces the effective data rate of a communication system.

In practice, the cellular and connectivity communication subsystems considered in this paper do not require effective data rates faster than 50 Mbps, most are even well below this level. A similar analysis for state of the art removable storage devices for mobile terminals leads to communication requirements around 100 Mbps.

As a result, it is proposed to use a shared serial bus system to connect the various communication pipe subsystems to the application engine. This approach is further supported by the fact that practical use cases will not require to operate all communication pipes in parallel. The requirements for low latency and on time delivery can be handled by choosing an appropriately high data rate for the shared serial link and by implementing a priority scheme.

C. Peripherals and Input/Output Devices

The least demanding requirements are imposed by low level peripherals and input/output devices. Examples for devices in this category include keyboard, ringer, speaker, microphone,

touchscreen, sensors, static displays, and others. Even though there will be real-time requirements for these devices, they can be satisfied easily, because they require low data throughput.

D. Implementation of Serial Links

The effective data rates required by the various subsystems discussed in the previous sections are summarized in fig. 5. As for the implementation of the serial links, two main candidate areas can be distinguished. The high speed, point to point serial links proposed for the display and camera subsystems should employ low voltage differential synchronous signaling techniques (LVDS) to ensure low levels of interference, robustness, and low power consumption. LVDS technology has been demonstrated to perform well even past the 1 Gbps mark. Typical voltage swings are in the range of 150 mV differential, and two pairs of wires are needed for each direction to carry data and clock signals. Power consumption has been estimated to be around 2 mW per signal pair.

The second candidate employs the more traditional single ended CMOS logic signaling to implement the serial link for communication pipes and low data rate peripherals.

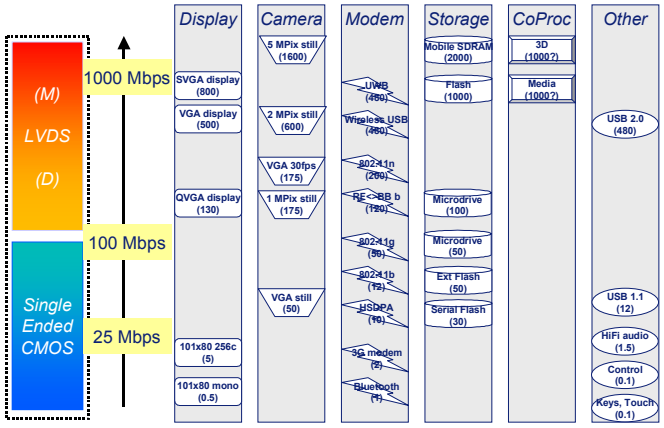

Fig. 5: Summary of the required data rates associated with the various subsystems. Some subsystems are included for reference only.

Dedicated protocols at physical (PHY) and medium access layer (MAC) are required to ensure smooth operation of the serial links under practical use cases and operating conditions. The description of these protocols is beyond the scope of this paper. Fig. 6 shows a summary of the interconnect topology used in the proposed architecture. The serial link used to connect the low data rate peripherals is not shown for clarity.

VI. CONNECTIVITY SUBSYSTEMS

We already explained in the introduction, why there is a trend that mobile terminals will have to provide also connectivity to networks other than cellular networks. We can separate this connectivity into broadcasting, networking, and storage connectivity. Examples for applications that may use this type of connectivity include mobile television, listening to broadcasting radio, wireless VoIP communication and Internet browsing in hotspot areas, various forms of memory cards, cable replacement, and others.

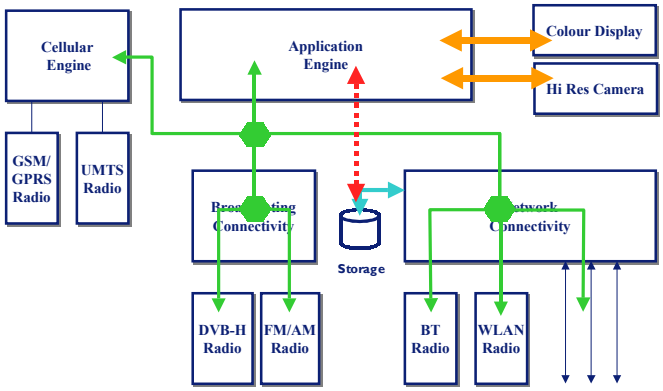

Fig. 6: The internal communications infrastructure is based on a network of digital serial links. Dedicated point to point links are used for the most demanding devices, while shared serial buses are used for medium and low data rate devices (not shown for clarity).

It is anticipated that longer term mobile terminals will have to support an increasing number of such connectivity technologies. On the networking side, consider Bluetooth, wireless LAN, Universal Serial Bus (USB), Near Field Communication (NFC), wired Ethernet. Future technologies may include wireless broadband access (IEEE 802.16) and ultra wideband technology (IEEE 802.15.3a). On the broadcasting side, analogue AM/FM receiver subsystems have been integrated with mobile terminals before. In the future, adding also capabilities to receive digital video and digital audio broadcast signals (DVB-H, DAB) will further expand the use cases supported by the terminal. Positioning information delivered by a GPS (Global Positioning System) module may also become important in the future and is already a requirement in some countries in the context of location information for emergency calls.

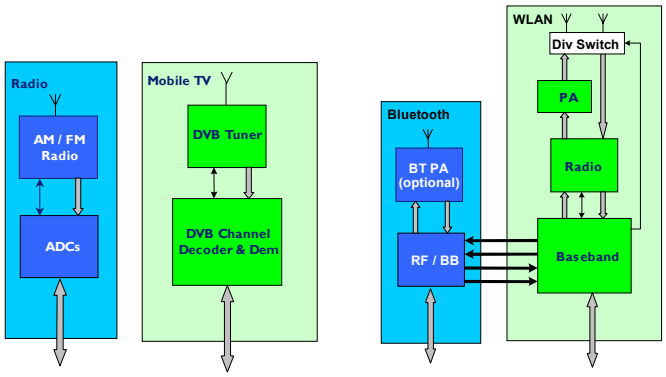

Fig. 7: A modular approach is chosen for the connectivity subsystems. The Bluetooth and Wireless LAN subsystems are tightly coupled to implement an effective solution for co-existence between these two subsystems.

A. Modular Approach

In order to maximize flexibility for the system integrator, a modular approach for each of the connectivity subsystems may be helpful. That way, a number of different products can quickly be implemented from a given set of available

connectivity subsystems. The subsystems will normally always implement the required physical layer functionality; the level of completeness of the MAC layer implementation can vary with the subsystem under consideration. An example is shown in fig. 7. The subsystems are connected to the application engine through the dedicated serial links discussed earlier.

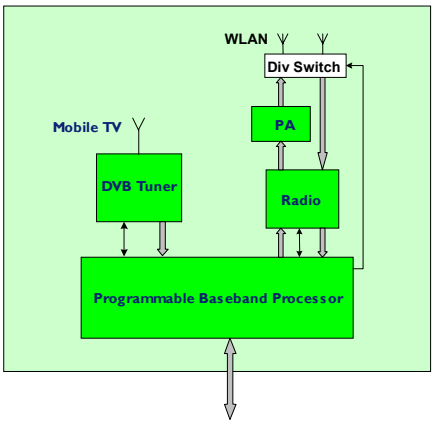

Fig. 8: Example for convergence between Digital Video Broadcasting (DVB-H) and Wireless LAN 802.11g subsystems using a programmable baseband processor.

The modular approach is also helpful, when it comes to software design. Each of the independent connectivity subsystems features its own microprocessor, which executes the protocol processing software up to the level defined by the overall software architecture. This allows for a loosely coupled system design, and the real-time critical tasks specific to a particular subsystem are directly handled by the associated microprocessor. This not only helps to offload the application engine, but makes overall system performance also more predictable.

B. Trend to further Convergence

On the long run, it can be expected that system integrators will not be able to efficiently implement the ever increasing number of connectivity alternatives without switching to more programmable structures. This fact is further emphasized by the realization that only a small fraction of the supported standards will be used for any particular use case at any one time; the other subsystems stay in idle mode or are completely turned off. As a result, in any given use case, a sizable share of the available hardware and compute power is not used, and a more cost efficient solution should be possible.

Two trends can be outlined, which point the way towards a possible solution. On the one hand, the processing power of programmable digital signal processors for a given power consumption has reached a level, where there is not much difference compared to a dedicated, fully hardware based solution. On the other hand, the technologies used by upcoming standards are more similar and as a result, the same hardware approach can be used to implement several standards with the same hardware.

As an example for this kind of convergence of standards we are showing, how the subsystems for wireless LAN and digital video broadcast can be handled by a single, programmable solution, see fig. 8. Both standards employ orthogonal frequency division multiplex (OFDM) schemes to carry the data over the radio channel. While initially the two standards were covered by independent subsystems, we have recently demonstrated that the same programmable hardware can be dynamically switched to support baseband processing for either of the two standards. The programmable baseband hardware connects to two radio subsystems covering the corresponding frequency ranges, the analogue interface signals are connected to the converters at the input of the baseband processing.

VII. SUMMARY

An architecture for a multi-media enabled mobile terminal has been proposed. It is based on a dual processor architecture separating the processing units for the cellular subsystem and its protocol stack processing from the application engine, which focuses on audio/video processing and user interface functions. Three different digital serial links are used to implement the internal communication infrastructure, they are designed to match the communication requirements of the associated subsystems. An example was given for convergence of connectivity subsystems, which will be needed to cope with the increasing number of different connectivity technologies that a mobile terminal will have to cover in the future.

ACKNOWLEDGMENT

The authors wish to acknowledge the contributions to this paper made by their colleagues Marcel Fuhren, Jean-Bernard Theeten, and Peter van den Hamer.

REFERENCES

[1] 3GPP, Technical Specification TS 27.010 “Terminal Equipment to User Equipment (TE-UE) Multiplexer Protocol”, www.3gpp.org

[2] Mobile Industry Processor Interface, www.mipi.org

[3] ARM Ltd., ARM926 Technical Documentation, www.arm.com

[4] Bluetooth Special Interest Group, www.bluetooth.org

[5] SATA Organization, www.sata.org

[6] IEEE 802.11 and 802.3 standards, IEEE Standards Association

[7] USB Implementer’s Forum, www.usb.org

Stefan Drude (M'95) received his Dipl.-Ing. degree in electrical engineering from RWTH Technical University Aachen, Germany, in 1983. The same year he joined Philips Semiconductors in Germany, where he was engaged in development of microcontroller software and radio paging systems. From 1994 to 1999 he has been based in Sunnyvale, CA, where he worked as manager of the system development team for mobile communications. This team was in charge of integrated circuits for cellular phones for the North American market. Under his responsibility, work has been done on IS-136 TDMA, IS-95 CDMA and IEEE 802.11 WLAN systems, and a number of successful products were developed and introduced to the market. In 1999, he moved back to Europe and became department head and program manager for wireless connectivity at Philips Semiconductors' Systems Laboratory Eindhoven, The Netherlands. In this function he was in charge of managing the system development and product architecture definition activities for Bluetooth and Wireless LAN chipsets. Since 2003, he has a new role as senior system architect for wireless connectivity in Philips Semiconductors' Advanced Systems Laboratory, where he is focusing on new wireless technologies for connected and cooperative systems and contributing to the definition of system solutions for future mobile terminals and wireless technologies for connected consumer applications.

Manfred Atorf received his Dipl.-Ing. degree in electrical engineering from Ruhr-University Bochum, Germany, in 1980. The same year he joined Philips Apparate Fabrik Wetzlar in Germany, where he has worked on the design of mixed signal circuits for car radio applications. After some international assignments as project leader he took over the group leader function for the electrical development group in 1985 and three years later for the car radio development department.. From 1990 onwards, he has managed Philips Car Systems' Wetzlar Development activity, responsible for the development of car radios and navigation systems. In 1993 he joint Philips Kommunikations Industrie, Nürnberg, Germany managing the development team for mobile phones. In 1994 he has built up Philips Semiconductors' Nürnberg activity as a System House focusing on the system development for digital mobile and cordless phones. During this period, he participated with his team in the development of the first Philips GSM handsets and DECT phones, as well as the next generations chipsets and complete system solutions. In 2001 he took over the management of the Competence Center Software mainly a concentration of crucial software resources across the organization with same focus as above. Since end of 2003 he is managing the Software Innovation Center being part ot the Business Development Group in the Communication Cluster of Philips Semiconductors in charge of the software architecture definition to support open operating system based future convergence platforms combining Multimedia, connectivity and telecommunicatoins for mobile portable devices.

Laurent Chivallier is Business Development and Strategy Manager with the Communications Cluster of Philips Semiconductors. He is in charge of the definition and deployment of Open OS strategy within PS in order to define future convergence platforms combining Multimedia, connectivity and telecommunications for mobile portable devices. From 1999 until 2003 Laurent held the position of Global Key Account Manager of a major cellular Japanese manufacturer with a worldwide responsibility. From 1997 to 1999 Laurent worked with VLSI Technology Inc. based in Paris where he was Field Marketing Manager Europe for VLSI One chip solutions of communications products GSM and DECT. From 1994 to 1997, he worked with Philips Consumer Communications in Le Mans as Senior Product Marketing Manager for GSM terminals. He was responsible of the definition and launch of the first Philips GSM terminals product line. From 1993 to 1994, He worked with Philips Kommunications Industrie in Nuremberg Germany as GSM Product Manager. From 1991 to 1993, he was Network Management Product Manager for Public telecommunication and cellular networks for TRT – Philips Communication Systems based in Paris. Laurent started in 1989 with Matra Communication based in Bois d'Arcy near Paris as GSM terminals Product Manager. Laurent received his Engineering Degree in Electronics specialized in Telecommunications at Ecole Centrale d'Electonique in Paris in 1988. He holds a Master degree specialized in Industrial Marketing and Operations Management from Ecole de Management de Lyon - EM Lyon in France in 1990

Kenneth Currie received his B.Sc. degree in Physics in 1988 from Edinburgh University, Scotland. From 1988 to 1992 he was employed by Plessey Semiconductors (later GEC-Plessey Semiconductors) where he worked on the development of DSP IC's. In 1992 he moved to work for Music Semiconductors who were based in Eijgelshoven, the Netherlands, where he spent two years working on the development of speciality memory devices such as CAM's. In the autumn of 1994 he joined Philips Semiconductors where he has remained to the present day, following on from periods spent working in the U.S. and Switzerland, he now works in Nijmegen as development manager of a team responsible for developing Application Engine IC's and SW. His current research interests are in how to approach the development of integrated circuit and software architectures to best match the evolution of the cellular industry.