Power Aware HW/SW Partitioning for a DVB-H Receiver Module

Ioannis Koryfidis, Sorin Cotofana - *Delft University of Technology, The Netherlands*
Joep van Gassel - *Philips Research, Eindhoven, The Netherlands*

Abstract—**In this paper, we consider a DVB-H receiver module and investigate effective means to find the most appropriate partition between software and hardware space (FPGA), while taking into account power consumption, hardware area constraints, and the communication between the blocks of the partition. To find the best partition for the considered application we utilize a simplified version of simulated annealing. An ARM processor is used for the software implementations and a Stratix FPGA device for hardware implementations. The software power dissipation is estimated using Sim-Panalyzer, while the hardware power dissipation is estimated using the PowerPlay Power Analyzer. Based on those estimates the annealing methodology provides a DVB-H system partition, which leads to a reduction in power consumption of up to 35% at the cost of up to 1703 FPGA logic elements. Our findings are important as they provide the means for an energy effective practical realization of a DVB-H enabled mobile device.**

Keywords— **DVB-H, HW/SW Partitioning, Power Consumption, Simulated annealing**

I. Introduction

THE Digital Video Broadcasting (DVB) Project [1] was born aiming primarily to standardize the technology involved in the creation and distribution of digital television signals.

Following the evolution of broadcasting, new dynamics in the consumer electronics market were created. In the last decades, electronic devices got smaller, so that today they can even fit in someone's palm. The flourishing electronics market has been continuously penetrated by new mobile products and the broadcasting industry had no choice but to be involved. Digital Video Broadcasting - Handheld (DVB-H) [2] was introduced as a technical specification [3] for bringing broadcast services to handheld receivers in November 2004.

Power consumption has been a major bottleneck in the design of efficient mobile devices, since the technology involving the development of batteries with bigger lifetimes is advancing very slowly. The major power consumption sources in CMOS circuits are the ones related to frequent charging and discharging of the capacitive loads and the ones related to leakage currents of logic gates. Except from the power dissipated in the device's overall logic, the platform's CPU, memory and buses also dissipate power as a consequence of executing a software application program.

Recently, Philips introduced its next generation DVB-H front-end System-in-Package solution, BGT215 [4]. Philips aims to differentiate from the competition by reducing the overall power consumption of a DVB-H enabled mobile device. In this line of reasoning we investigate in this paper effective means to reduce the power consumption of a reference system consisting of a DVB-H module and an ARM-based application processor.

The main idea behind our approach is to extract portions of the software running application as potential candidates for hardware realization, where they would consume less power. This type of strategy is called hardware/software partitioning and it can be applied in any level of design abstraction, which in our case is the system level.

The main partitioning technique that we primarily envisage is to split the DVB-H protocol stack functionality into basic building blocks according to a chosen granularity and estimate the power consumed both for software (using Sim-Panalyzer [5], a power simulator for the ARM) and hardware implementations (using Altera's PowerPlay Power Analyzer [6]) of each and every block. Then, we can identify which hardware/software partition provides the best power consumption reduction over an all-software solution, using an efficient method that also considers hardware area as a constraint. Moreover, a significant factor that must not be omitted from our estimations is the communication between hardware and software.

As a partitioning methodology, Simulated Annealing [7], a common method used in solving combinatorial optimization problems, is utilized. The employed algorithm provides us with the overall power consumption values and hardware areas for the best partitioning scenarios, according to a predefined cost function.

Our experimental results indicate that, utilizing HW/SW partitioning with Simulated Annealing, power reductions of up to 35% in comparison to the all-software solution can be achieved, at a cost of up to 1703 FPGA Logic Elements. Moreover, by comparing the Simulated Annealing results with the ones generated via exhaustive search we conclude that the Simulated Annealing algorithm is correctly choosing between nearby solutions, according to the considered cost function.

This paper is organized as follows. Chapter 2 introduces the system's specifications providing the necessary background knowledge on DVB-H. Chapter 3 presents the power estimation methodology for the hardware and software implementations and for the hardware/software communication. Chapter 4 explains the partitioning methodology that we utilize. Chapter 5 presents the experimental results and Chapter 6 concludes this paper with a summary of our findings, and provides possible directions for relevant future research.

II. DVB-H

The MPEG-2 Transport Stream (TS) is the means of transmitting information in the DVB standards. The way of carrying IP datagrams in an MPEG-2 TS is realized using Multiprotocol Encapsulation (MPE). Each IP data-

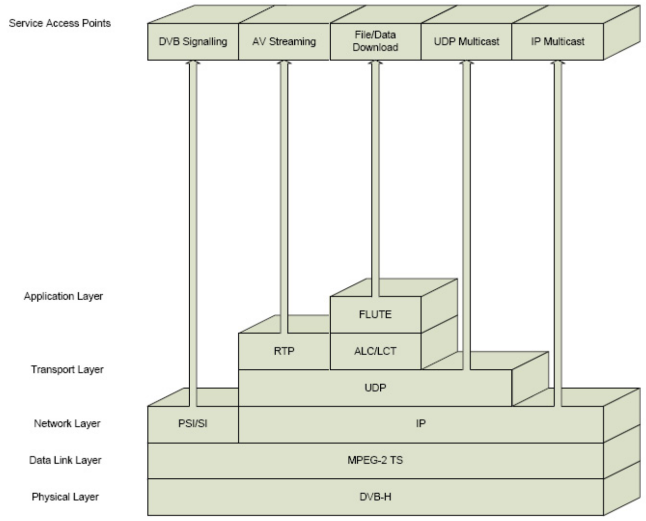

Fig. 1. Protocol Hierarchy for DVB-H IPDC

gram is encapsulated into one MPE section. An Elementary Stream (ES) is then formed out of MPE sections with a particular program identifier (PID).

IP datacast (IPDC) [8] refers to the set of specifications created in order to fill the gap between independent networks. With IP datacast it is possible to have a single convergent terminal, which in the case of DVB-H is a mobile device, that consumes a variety of broadcast content and services, with full compatibility of the networks involved.

Real-time content, such as real-time audiovisual content (H.264 video and AAC+ audio), is delivered via Real-Time Protocol (RTP), while non-real-time content is delivered via File Delivery over Unidirectional Transport / Asynchronous Layered Coding (FLUTE/ALC) data carousel.

The H.264/AVC specification introduces the Video Coding Layer (VLC) and the Network Abstraction Layer (NAL) as the basic elements describing video content. VCL is responsible for the video features of H.264 while NAL is responsible for the formatting of VCL data into NAL units, appropriate for transport over the utilized network. The transport of High Efficiency AAC v2 audio is realized through Access Units (AUs) that reside inside RTP packets. The AU payload contains the data of one AAC frame. Finally, FLUTE resides over the Asynchronous Layered Coding (ALC) protocol instantiation. ALC consists of a Layered Coding Transport (LCT) block and a Forward Error Correction (FEC) block. ALC uses the LCT building block to provide in-band session management functionality.

Figure 1 shows the protocol stack of the IP Datacast transmission system.

The DVB-H system that we wish to optimize in terms of power dissipation is realized both in software (C++) and in hardware (VHDL). A brief list of requirements is the following:

1. The reference software application is running on an ARM processor.
2. The hardware implementation resides in an FPGA device.
3. The communication between software and hardware modules is realized through a system bus.
4. The partitioning methodology should be applied in the functional area between the output of the PID filter and the input of the H.264, AAC+ and FLUTE decoding blocks. MPE-FEC and PSI/SI extraction are not examined.

In order to optimize power dissipation, our plan is to initially partition the system's functionality in distinct parts according to a chosen granularity and estimate the power dissipation for each part both for the software and the hardware implementation. These estimations are input vectors to our partitioning methodology, which can calculate efficiently the most appropriate partitioning scenarios for the power optimization problem.

In our case, examining the desired system's functionality, which is depicted in Figure 1, we chose to use the protocol layers of IP Datacast over DVB-H as building blocks. This choice defines the granularity, which belongs to a high-level of abstraction. The choice originates from the system's behavior, which consists from successive fragmentations of an initial MPEG-2 Transport Stream passing from the Data Link, Network, Transport and Application Layers.

The functional partition of the envisioned DVB-H system in basic elements, according to the chosen granularity and requirements is presented in Figure 2. The blocks in dark color represent the blocks of interest, i.e., potential candidates for hardware implementation, while the blocks in light color represent the corresponding neighboring blocks. Moreover, the lines connecting the blocks represent the corresponding communication.

An MPEG-2 Transport Stream contains a collection of Elementary Streams of different kinds of data. In our system we are examining real-time video and audio and non-real-time file deliverables. If we envisage in our hardware system that each of the above blocks exists only one time in the system, i.e., there is only one Section filter, one IP filter etc., then there are three distinct paths in order to acquire specific content (video, audio or file) at a time and only one of the paths is activated. The three paths are the following:

1. *H.264 extraction*
Section filter → IP filter → UDP filter → RTP filter → NAL filter
2. *AAC+ extraction*
Section filter → IP filter → UDP filter → RTP filter → AAC filter
3. *FLUTE extraction*
Section filter → IP filter → UDP filter → LCT filter

This distinction is very important because the partitioning solutions for all three paths must converge until the UDP filter, since it is common in every path. Moreover, convergence has to exist in the RTP filter for the video and audio paths.

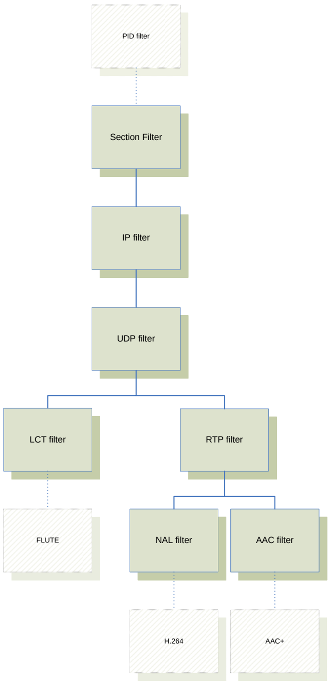

Fig. 2. DVB-H Functional Partitioning

III. Power Estimation

A. Hardware

For the realization of the hardware of the additional modules (from Section filter to NAL, AAC, and LCT filter) we used the Altera's Stratix EP1S10 FPGA [9] as target device. For the estimation of the power dissipation of the building blocks depicted in Figure 2 as potential candidates for hardware implementation, the Quartus II PowerPlay Power Analyzer [6] is used. Power Analyzer requires synthesis of the design targeting a specific device and a collective signal activity file to be used for toggle rate extraction. Synthesis is realized using the Quartus II software. Using the synthesis tool of Quartus II we also extract the hardware area that each building block requires in order to be synthesized, which will be used as an input to our partitioning methodology.

In order to evaluate the power dissipation for each individual block in Figure 2, a reference design consisting of all the modules properly interconnected was created. Then, applying separate MPEG-2 Transport Streams that contained data for H.264 video, AAC+ audio or FLUTE as an input, the power dissipation for each block was estimated by the Quartus II, through hierarchical estimations that are supported by the Power Analyzer. These MPEG-2 Transport Streams were taken directly from a file that was generated by a DVB-H encapsulator.

B. Software

In order to estimate the power dissipated by the software executed on the ARM processor, we use Sim-Panalyzer [5], an augmentation to the SimpleScalar [10] performance simulator. ARM binaries were produced using an ARM-Linux cross-compiler.

The Sim-Panalyzer tool is an infrastructure for microarchitectural power simulation. It is positioned above the "sim-outorder" component of the SimpleScalar simulator. The Sim-Panalyzer program contains components that model specific parts of the ARM processor. Sim-Panalyzer focuses efficiently on basic microarchitectural blocks and provides power information over a wide range of power dissipation sources, such as caches, clock trees, external I/O, on-chip memories, and datapath and execution blocks.

C. Communication between HW-SW

In our methodology, we take into account the communication cost between two building blocks only when they reside in different implementations. We, therefore, assume the communication between hardware-hardware or software-software negligible. In order to estimate the power dissipation for the transfer of a specific amount of bytes from hardware to software and vice versa, we assume that power is dissipated only in the bus connecting the processor (in our case, ARM) and the reconfigurable logic (in our case, Stratix FPGA). In this line of reasoning, we use the models presented in [11] for the power dissipated in the interconnect between data path and external memory in an embedded system. In cases where software is sending data to hardware or hardware is sending data to software, the receiver of data can be seen as a memory component for the transmitter, so this model provides a good approximation of the actual power dissipation.

The power dissipated by a specific bus b, P_{bus} in mW is defined [11] as:

$$P_{bus} = W_{bus} \cdot a_{toggle} \cdot f_{access} \cdot (C_{driver} + C_{load}) \cdot V_{dd}^2 \cdot 0.001 \quad (1)$$

where:

W_{bus} is the width of bus b in bits,
a_{toggle} is the toggle rate of bus data,
f_{access} is the operating frequency of the bus in MHz,
C_{driver} is the capacitance of the buffer that drives bus b,
C_{load} is the total load capacitance of bus b, in pF,
V_{dd} is the bus power supply, in Volts and,
0.001 is a factor included to obtain the power in mW.

Using Equation (1), we evaluated the per block power dissipation estimates and they are presented in Table I. Note that there exist more than one estimations for the outputs of the UDP and RTP filters, since they may participate in more than one path.

IV. HW/SW Partitioning

Simulated annealing (SA) [7] is a generic probabilistic meta-algorithm for global optimization problems. It locates a good approximation of the global optimum for a given function in a large search space. The primary goal in our case is to use SA in order to find an area-satisfying partition $Par_1 = \{H, S\}$ so that power consumption $P = min$.

TABLE I
HW/SW Communication Power

Block	a_{toggle}	HW/SW Comm. (mW)
Section filter	18.13%	34.49
IP filter	16.46%	31.31
UDP filter (NAL)	11.32%	21.54
UDP filter (AAC)	5.01%	9.53
UDP filter (LCT)	10.92%	20.77
RTP filter (NAL)	10.57%	20.11
RTP filter (AAC)	4.29%	8.16
NAL filter	10.43%	19.84
AAC filter	4.21%	8.01
LCT filter	7.63%	14.51

First of all, the metrics for the cost function in our system are the hardware area in Logic Elements and the power dissipation in mW. In HW/SW partitioning a random perturbation of a configuration implies a move of a single vertex (see Figure 2) from HW to SW or from SW to HW. This move alters the values in the edges between the neighboring vertexes, and the communication costs change together with the value of the moved vertex. Given that we cannot really assume that a move of a vertex from SW to HW simply reduces the system's power consumption, because the resulting communication cost overhead may be higher than the power consumption gain (generally, hardware dissipates less energy than software). Nonetheless, this move definitely increases hardware area, as it is an independent factor. The effects of a move of a vertex from HW to SW can be discussed in a similar way.

In the main iteration of the SA algorithm, the cost function corresponding to a potential move is evaluated. Simulated annealing inherently uses randomization to overcome local minima and moves that do not improve the optimization objective are accepted with some probability.

The hardware area and power consumption values in our cost function have to be normalized since they refer to different units. Normalization of hardware area is achieved by dividing the hardware area with the hardware area constraint. For the power consumption, normalization is achieved by dividing the power consumption with the power consumption constraint. Instead of using the simplest scheme of just adding two normalized values to evaluate the cost function, we use a modified version of [12], where there is a penalty for moves that are crossing the constraint boundaries. Penalties are proportional to the distance from the boundary constraint. Moves causing boundary violations in both power consumption and hardware area are severely penalized. Moreover, since we aim to optimize power consumption, a 2x weight to the power consumption factor is given.

Based on the above discussion, our cost function is algorithmically described as:

```
$cost\_func \Leftarrow 2\times(cur\_pow)/(pow\_bound)+(cur\_area)/(area\_bound)$
if move causes only power boundary violation then
    $cost\_func \Leftarrow (cost\_func) \times [(cur\_pow)/(power\_bound)]$
else if move causes only area boundary violation then
    $cost\_func \Leftarrow (cost\_func) \times [(cur\_area)/(area\_bound)]$
else if move causes area boundary AND power boundary violation then
    $cost\_func \Leftarrow (cost\_func) \times 100$
end if
```

V. Experimental Results

Initially, we present the hardware area required in the FPGA for the implementation of each of the blocks in Figure 2. These values originate from the synthesis of the implemented design and are shown in Table II.

Block	Hardware Area (LEs)
Section Filter	571
IP Filter	532
UDP Filter	127
RTP Filter	165
NAL Filter	232
AAC Filter	51
LCT Filter	25

TABLE II
Hardware design area values

In Tables III to V, the power dissipation values for hardware and software implementations of the H.264 extraction, the AAC+ extraction, and the FLUTE extraction path are presented, respectively. We note that the static thermal power dissipation for the EP1S10 Stratix FPGA device is *187.5 mW*.

Block	HW Power (mW)	SW Power (mW)
Section Filter	9.29	83.00
IP Filter	8.63	79.20
UDP Filter	8.19	77.40
RTP Filter	8.50	71.70
NAL Filter	6.20	60.60

TABLE III
Power dissipation values for the H.264 path

Block	HW Power (mW)	SW Power (mW)
Section Filter	9.29	83.00
IP Filter	8.63	79.20
UDP Filter	8.02	62.50
RTP Filter	8.11	59.20
AAC Filter	5.67	58.90

TABLE IV
Power dissipation values for the AAC+ path

We next introduce in Table VI the total power dissipation that were derived from Tables III to V for each of the three paths, when an all-hardware and an all-software solution are realized.

In order to have power dissipation reduction over an all-software solution, we may only explore the partitioning pos-

Block	HW Power (mW)	SW Power (mW)
Section Filter	8.60	78.40
IP Filter	7.63	70.90
UDP Filter	7.72	68.00
LCT Filter	4.62	64.10

TABLE V
Power dissipation values for the FLUTE path

Path	Total HW Power (mW)	Total SW Power (mW)
H.264	228.31	390.30
AAC	227.22	361.20
Flute	216.07	299.80

TABLE VI
Total Power Dissipation for all-HW and all-SW implementations

sibilities that do not surpass the all-software power dissipation presented in Table VI. However, the fixed power dissipation cost that is required in order to integrate the EP1S10 Stratix FPGA to our system is the FPGA's static power dissipation (187.5 mW). So, taking into account this mandatory power component, the maximum allowed power dissipations for our partitioning scenarios are derived from the difference between the power dissipations of an all-software solution and the FPGA's static power. These values are depicted in Table VII.

Path	Max. allowed Power (mW)
H.264	202.8
AAC	173.7
Flute	112.3

TABLE VII
Maximum allowed Power Dissipations in order to have Power Reduction

The values in Table VII, the hardware and software power estimation values in Table VI, and the HW/SW communication values in Table I are the inputs to the Simulated Annealing algorithm. Based on them the SA can derive the best partitioning scenario for each individual path. Afterwards, we explore the possibilities of further power dissipation reduction by decreasing the limits of Table VII. The hardware area constraints are set to the values of Table II, since our algorithm will search for the solutions that are as below these constraints as possible. Tables VIII, IX, and X present the outputs of our Simulated Annealing algorithm for the corresponding paths.

The partitioning scenarios for all the paths, according to Tables VIII, IX, and X are presented in Table XI.

Power Boundary (mW)	182.6	162.2	141.9	min
Actual Power (mW)	177.18	149.01	115.32	40.81
HW Area (LEs)	1095	1056	1395	1627
Power reduction (mW)	25.62	53.79	87.48	161.99
Power reduction (%)	6.56	13.78	22.41	41.50
Par (H.264-Table XI)	a	b	c	d

TABLE VIII
Results of the SA algorithm for the H.264 path

Power Boundary (mW)	156.3	138.9	121.6	min
Actual Power (mW)	147.92	135.04	101.11	39.72
HW Area (LEs)	875	1319	1395	1446
Power reduction (mW)	25.78	38.66	72.59	133.98
Power reduction (%)	7.14	10.70	20.09	37.09
Par (AAC-Table XI)	a	b	c	d

TABLE IX
Results of the SA algorithm for the AAC+ path

The partitions for every path must converge in implementation for the Section Filter, the IP Filter, and the UDP Filter. The only partitioning scenario that offers that, except the one of an all-hardware solution that we call PS_0, is the one consisting of H.264(c), AAC+(c) and FLUTE(a) or FLUTE(b) in Table XI. We call this partition set PS_1 in the case of FLUTE(b) and PS_2 in case of FLUTE(a). However, we observe that the H.264 and AAC+ paths, which are dominating regarding power consumption, converge as well in H.264(b), AAC+(a). We investigate the power reductions also for the last case, even though the {SW, HW, HW, HW} partition of FLUTE (consuming 131.91 mW and taking up 684 LEs) is not included in the results obtained from the SA algorithm. We call this partition set PS_3.

Table XII presents a comparison between the aforementioned partition sets. Note that the negative sign of the power reduction for FLUTE in PS_3 means that there is an increase in power dissipation in comparison with an all-software solution. The hardware area values correspond to the hardware implemented building blocks of all three paths. Note that (v), (a) and (f) correspond to the H.264, AAC+, and FLUTE paths, respectively.

The amount of possible partitioning scenarios p depends on the number of building blocks n under investigation as $p = 2^n$. The problem of finding the optimal partition is categorized as NP-complete, since it can be solved in exponential time only. In the general case the exact solution cannot be found as it requires exhaustive search and this is not feasible for large n. In our case however there are 32 partitions only for the H.264 and AAC+ paths and 16 for the FLUTE path. This allows us to perform an exhaustive search in order to evaluate how effective the SA heuristic is in in finding the most appropriate partition. In Figures 3, 4, and 5, the {hardware area, power consumption} pairs for every partition are depicted for the H.264, AAC+, and FLUTE path, respectively. The portion of the graphs that corresponds to pairs giving power reductions is highlighted and the proposed partitions from the Simulated Annealing algorithm are also presented. The letters below the proposed pairs correspond to the partitions of Table XI.

In Figure 3 we can identify the role of the Simulated

Power Boundary (mW)	110.1	89.84	78.61	min
Actual Power (mW)	108.82	28.57	28.57	28.57
HW Area (LEs)	1230	1255	1255	1255
Power reduction (mW)	3.48	83.73	83.73	83.73
Power reduction (%)	1.16	27.93	27.93	27.93
Par (Flute-Table XI)	a	b	b	b

TABLE X

Results of the SA algorithm for the FLUTE path

Path	Par	sf	ip	udp	rtp	nal	aac	lct
H.264	a	HW	SW	HW	HW	HW		
	b	SW	HW	HW	HW	HW		
	c	HW	HW	HW	HW	SW		
	d	HW	HW	HW	HW	HW		
AAC	a	SW	HW	HW	HW		HW	
	b	HW	HW	SW	HW		HW	
	c	HW	HW	HW	HW		SW	
	d	HW	HW	HW	HW		HW	
Flute	a	HW	HW	HW				SW
	b	HW	HW	HW				HW

TABLE XI

Partitioning Scenarios

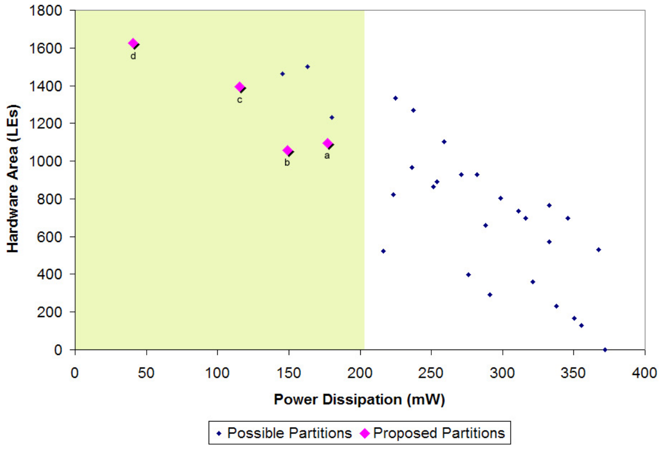

Fig. 3. Possible and Proposed Partitions for the H.264 path

Set	Area (LEs)	Reduction (%)	Avg Reduction (%)
PS_0	1703	41.50 (v) 37.09 (a) 27.93 (f)	35.51
PS_1	1420	22.41 (v) 20.09 (a) 27.93 (f)	23.48
PS_2	1395	22.41 (v) 20.09 (a) 1.16 (f)	14.55
PS_3	1132	13.78 (v) 7.14 (a) -6.54 (f)	4.79

TABLE XII

Comparison of Partitioning Scenarios

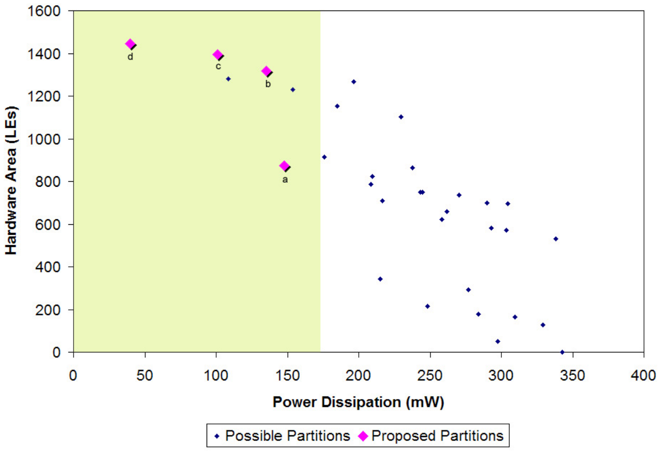

Fig. 4. Possible and Proposed Partitions for the AAC+ path

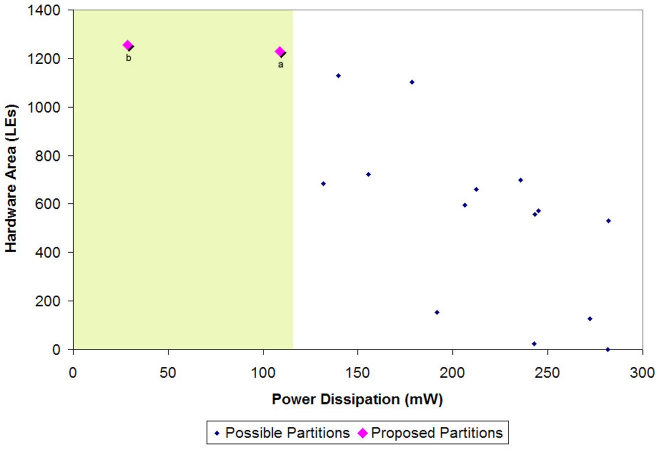

Fig. 5. Possible and Proposed Partitions for the FLUTE path

Annealing cost function in the choice of the best partition. The possible pair of values above point (a) is not chosen, since it provides much higher hardware area value for approximately the same power dissipation value as point (a). A similar situation occurs for point (b), which dominates two points with much higher hardware area values and close power dissipation values. Finally, points (c) and (d) the only possible choice for the Simulated Annealing algorithm for the corresponding constraints, since they do not have any other competitor points in their range.

In Figure 4 we observe again a similar case as that of point (a) in Figure 3 for the point above it. However, examining this graph makes it obvious that hardware area is not the only defining parameter regarding the selection between two close points. In the case of point (c), where the hardware area is higher than that of the point on the right of (c) but the power dissipation is lower, the Simulated Annealing algorithm selects (c) because the power dissipation has larger weight in comparison to hardware area in the cost function. This dictates that for points of so close values as it is in this case, the point corresponding to more hardware area - less power dissipation is chosen. Again, points (b) and (d) are the only possible choice, since they do not have any competitors in range.

Finally, in Figure 5 we identify only two solutions in the power reduction range. These two points have no competitors, so they are blindly chosen as in the previous figures. Point (a) is selected when the power dissipation boundary is close to the boundary of the power reduction range and point (b) is selected in all the other cases, which include power dissipation boundaries below that of point (a).

VI. Conclusion

In this paper, we investigated the possibility to find the most appropriate partition between the software and the hardware space for a DVB-H receiver module, while aiming to reduce the power consumption. To achieve such a partition, we proposed a power aware HW/SW partitioning methodology based on Simulated Annealing. The method can be applied for any application when the power dissipation values of hardware and software implementations, as well as the hardware areas and power dissipation values due to HW/SW communication are available. Our experimental results indicate that, by utilizing the HW/SW partitioning based on Simulated Annealing, power reductions of up to 35% in comparison to the all-software solution can be achieved, at a cost of up to 1703 FPGA Logic Elements, and that the Simulated Annealing algorithm is correctly choosing between nearby solutions, according to the cost function. Future research can include a modified methodology that would save the designer from time-costly power estimation procedures for hardware and software by only considering communication constraints between blocks. This should be in principle possible at the expense of some lost accuracy, since the communication between hardware and software space consumes relatively considerable amounts of power. Finally, an addition to the SA algorithm's functionality that would examine multiple paths with common building blocks may constitute another possible future research direction.

References

[1] Digital Video Broadcasting (DVB). [Online]. Available: http://www.dvb.org/

[2] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices," in *Proc. IEEE*, vol. 94, no. 1, Jan. 2006, pp. 194–209.

[3] *Transmission System for Handheld Terminals (DVB-H)*, ETSI Std. EN 302 304 v1.1.1, 2004.

[4] Philips DVB-H solution BGT215 (TV-on-mobile for Europe and Asian DVB-H markets). [Online]. Available: http://www.semiconductors.philips.com/acrobat/literature/9397/75015522.pdf

[5] Sim-Panalyzer, The SimpleScalar-Arm Power Modeling Project. [Online]. Available: http://www.eecs.umich.edu/p̃analyzer/

[6] Altera Corporation - Quartus II. [Online]. Available: http://www.altera.com/products/software/products/quartus2/qts-index.html

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] "IP Datacast over DVB-H: Architecture," DVB BlueBook A098, Nov. 2005.

[9] Altera Corporation - Stratix FPGAs. [Online]. Available: http://www.altera.com/products/devices/stratix/stx-index.jsp

[10] SimpleScalar Tool Set. [Online]. Available: http://www.simplescalar.com/

[11] F. Catthoor, E. d. Greef, and S. Suytack, *Custom Memory Management Methodology: Exploration of Memory Organisation for Embedded Multimedia System Design.* Kluwer Academic Publishers, 1998.

[12] S. Banerjee and N. Dutt, "Very fast Simulated Annealing for HW-SW partitioning," CECS, Tech. Rep., June 2004.