

## INFSCO-ICT-216203 DA VINCI

### D3.1.3 v1.0

#### *Power Estimation and global Asic 45nm results report disclosing: area, frequency, throughput, power consumption*

**Contractual Date of Delivery to the CEC:** 30 06 2010 (*as specified in the contract*)

**Actual Date of Delivery to the CEC:** 30 06 2010

**Author(s):** V. Heinrich, T.Simon

**Participant(s):** STM

**Workpackage:** WP3

**Estimated person months:** NA

**Security:** PU

**Nature:** R

**Version:** 1.0

**Total number of pages:** 15

**Abstract:** In this deliverable we explain the methodology used for the power estimation of the DaVinci decoder based on the gate netlist obtained in sub-task 3.1.1. Results of power figures in different scenarios are provided, taking into account active and inactive periods of the LDPC decoder in the potential application. We also recall the main results obtained during the design of the gate netlist of the DaVinci ASIC decoder.

**Keyword list:** Non-binary LDPC, ASIC implementation, power estimation, consumption

**Disclaimer:** N/A

---

## **Executive Summary**

Power consumption of a digital block such as LDPC decoder can be a concern, especially when it is intended to be used in battery-powered devices. In this deliverable, we give some basics about power estimation in the ASIC design flow and how we applied it to our DaVinci decoder.

The results of the power estimation in different scenarios are provided. We tried to simulate study cases as close as possible to real applications by taking into account active (decoding) and inactive periods of the LDPC decoder. The estimation has been done mainly in two representative modes (mode 9 and mode 15), i.e. code rate and frame length, but the results obtained can be extended to the general case.

**Authors**

<b>Partner</b>	<b>Name</b>	<b>Phone / Fax / e-mail</b>
<b>STMicroelectronics (STM)</b>		
	Vincent Heinrich	Phone: +33 (0)4 76 58 70 60 e-mail: <a href="mailto:vincent.heinrich@st.com">vincent.heinrich@st.com</a>
	Thierry Simon	Phone: +33 (0)4 76 92 58 41 e-mail: <a href="mailto:thierry.simon@st.com">thierry.simon@st.com</a>

---

## Table of Contents

<b>Executive Summary .....</b>	<b>2</b>
<b>Authors.....</b>	<b>3</b>
<b>Table of Contents .....</b>	<b>4</b>
<b>List of Acronyms and Abbreviations.....</b>	<b>5</b>
<b>1. Introduction .....</b>	<b>6</b>
<b>2. ASIC power estimation of DaVinci decoder.....</b>	<b>8</b>
2.1 Introduction to power estimation.....	8
2.2 Power estimation tool and methodology .....	9
2.2.1 PrimePower tool presentation .....	9
2.2.2 Power analysis flow .....	9
2.2.3 DaVinci scenarios .....	10
2.3 Power estimation results .....	11
2.3.1 Gate netlist properties .....	11
2.3.2 Measured power in active and passive periods .....	11
2.3.3 Mean number of iterations study.....	13
<b>3. Conclusion .....</b>	<b>15</b>

---

## List of Acronyms and Abbreviations

ASIC	Application Specific Integrated Circuit
BP	Belief Propagation (decoding algorithm)
CN	Check-node
DSP	Digital Signal Processing
EMS	Extended Min-Sum
FER	Frame Error Rate
GF	Galois Field
HLS	High-Level Synthesis
LDPC	Low Density Parity Check (codes)
LIFO	Type of memory “Last-In First-Out”
LLR	Log Likelihood Ratio
LUT	Look-Up Table
NB	Non Binary
PE	Power Estimation
PnR	Place and Route
RAM	Random Access Memory
RTL	Register Transfer Level
SNR	Signal-to-Noise Ratio
SoC	System on Chip
VN	Variable-node
XOR	Exclusive OR gate

## 1. Introduction

Task 3.1 is entitled: “Demonstration of DaVinci codec in 45nm technology”. It is divided in three sub-tasks related to the design of a chip, from the high level description (algorithm) to the placement and routing step. These sub-tasks, as specified in the contract, are described below and a graphical representation of task 3.1 is given in Figure 1.

### Subtask 3.1.1 : Build Channel Coding/Decoding GATE description.

This action, focusing on physical design and implementation, will start from the specification-oriented results obtained in WP6. It is clear that the subtasks 3.1.1 and 3.1.2 are closely related and will need many loops to converge. Timing closure is expected to be uneasy, and will have to include some exploration due to buffer addition. Concerning the degree of parallelism, some trade-offs will have to be found between speed (many similar operators) and area (few similar operators). To prevent conflicts in RAM access, measures will have to be taken involving sequential tuning and/or hardware duplication.

### Subtask 3.1.2 : Placement & Routing benchmarking.

Memory mapping and Place & Route will also require an in-depth exploration. Logic routing is a challenge. This will require the definition of new memory access architectures so as to extract several hundreds of data per clock cycle. Preliminary experiments show that logic only fill 20% of the critical part of such circuits, while the rest is devoted to routing. For ASIC implementation, feasibility demonstration will be considered as successful if routing benchmarking shows that overflowed routing edges are less than 1 over 1000 of the routing edges.

### Subtask 3.1.3 : Estimation of the Power consumption of the solution using Power estimation tools.

Power estimations using gate characterizations will be performed. The static power consumption (leakage) and dynamic power consumption in different scenarios will be estimated.

The first sub-task is the most time consuming as it implies architectural explorations, hardware design and logic synthesis, all these steps including a validation process in order to ensure that the reference model is actually implemented. The final result is a gate netlist, that is to say a description based on standard cells linked together by wires. It is used to obtain accurate timing and area information as requested in sub-task 3.1.1.

In addition, the gate netlist becomes the input for sub-tasks 3.1.2 and 3.1.3, whose goal is to provide routing feasibility and power consumption information. The results concerning these tasks are described in report D3.1.2 and D3.1.3 respectively.

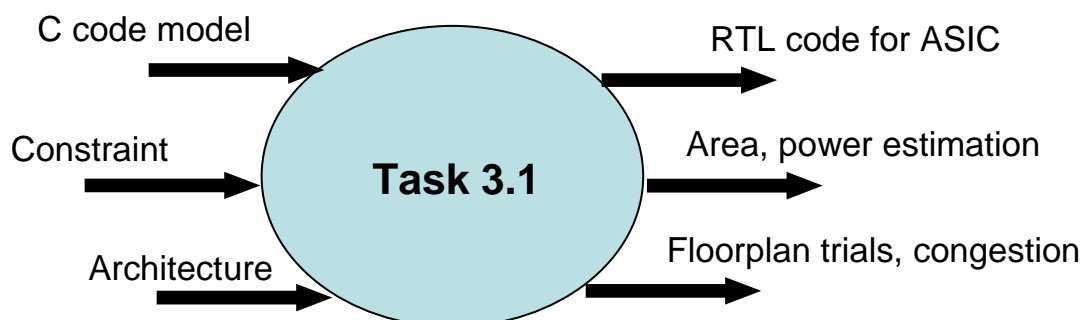


Figure 1 : Task 3.1 graphical representation

This report concerns task 3.1.3 which focuses on the power consumption evaluation. This metric becomes more and more critical, as the transistors density increases. The first issue is the heat dissipation which is limited and proportional to the package size. High power dissipation may require expensive specific packages and techniques. The second point concerns the battery-powered devices: a high autonomy is a differentiator and is directly linked to the power consumption of the circuit. These are the reasons why power consumption evaluation has been carried out within DaVinci project.

In a first section we introduce the concept of power estimation. We present the tool used for this and the place of power estimation in the design flow. Then we present the different assumptions and scenarios applied to the power estimation in this project. After having explained how to calculate the mean power of a frame decoding, the results are presented in different representative conditions. We finally comment and conclude on the power figures obtained.

## 2. ASIC power estimation of DaVinci decoder

### 2.1 Introduction to power estimation

Power estimation (PE) can be done at several stages of the design flow applied to a circuit. Nevertheless, the later in the flow the power estimation is performed, the most accurate it is but also the slower it is. Even at the system level, a first evaluation can be done by evaluating the number of operations (additions, multiplications, ...) that will be needed per second. Today, recent tools are able to give good power estimation at RTL level, by running a simulation based on realistic stimuli.

In our case, we have performed the power estimation at the gate level, i.e. after the logic synthesis. This model allows getting accurate power figures, under the assumption that relevant stimuli are provided to the simulator. Thanks to the gate netlist description, the power characteristics of each standard-cell implemented is well known. The remaining uncertainties are due to the power dissipated in the wire lines and especially in the clock tree. The clock tree is responsible for an important part of the dynamic power consumption. For this reason PE tools using gate netlist description take into account an estimation of the power dissipated in the wires and in the clock tree.

The power consumption of digital CMOS circuits is generally considered in terms of three components:

- The *dynamic power* component, related to the charging and discharging of the load capacitance at the gate output. Dynamic power is the power dissipated when the circuit is active. A circuit is active anytime the voltage on a net changes due to some stimuli applied to the circuit
- The *short-circuit power* component. During the transition of the input line from one voltage level to another, there is a period of time when both the PMOS and the NMOS transistors are on, thus creating a path from  $V_{DD}$  to ground. This component is generally integrated in the dynamic power.
- The *static power* component, due to leakage, that is present even when the circuit is not switching, that is, when it is inactive or static. This, in turn, is composed of two components - *gate to source leakage*, which is leakage directly through the gate insulator, mostly by tunnelling, and *source-drain leakage* attributed to both tunnelling and sub-threshold conduction.

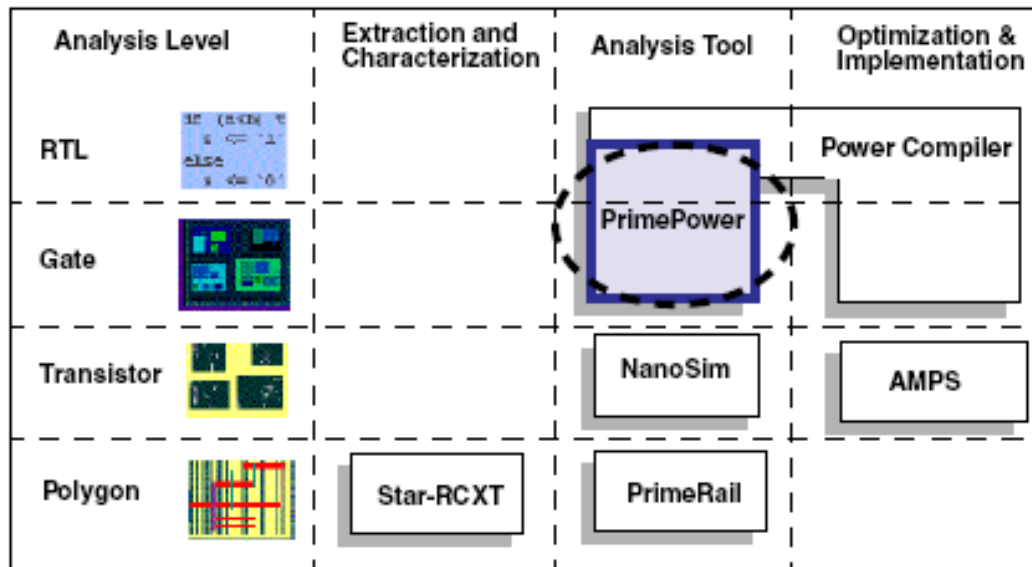
The leakage power is generally lower than the dynamic power, depending on the application. However, when considering the real use of a digital block, the power budget may become essentially due to the static power for some applications. As an example, a TV remote controller is dynamically used only when the user wants to change the channel, which represents a very small duration compared to the total battery life.

Concerning DaVinci, assuming that a non-binary LDPC is used in a cell phone baseband, the static power will become significant as the transceiver will be used only when communicating. Thus, in order to limit the leakage power within DaVinci technology, we decided to use a specific low-power 45nm technology. The standard-cells associated are slower but they have a greatly reduced leakage power.

## 2.2 Power estimation tool and methodology

### 2.2.1 PrimePower tool presentation

In order to keep the best tool compatibility and reduce the risks, we used a tool from Synopsys for the power estimation. This tool, known as PrimePower, is easily integrated in the design flow, after the logic synthesis. “*PrimePower is a gate-level power analysis tool that accurately analyzes power dissipation of cell-based designs. It is intended as an advanced solution for ASIC and structured custom circuit designers who are developing products for power-critical applications such as portable computing and telecommunications.*”



**Figure 2 : PrimePower and other Synopsys tools**

The position of PrimePower and other CAD tools from Synopsys in the flow can be seen in Figure 2. PowerCompiler can help for power optimizations and is closely related to PrimePower; however we did not try particular power optimizations in this project.

PrimePower performs the following steps to accurately analyze the power of the design:

1. Based on the design connectivity, design constraints, and wire capacitance, PrimePower determines the transition times for all the pins within the design.
2. For average power analysis, PrimePower determines the state and path-dependent switching for all the nodes within the design. If nets are not annotated, the propagation engine is used to determine the switching of unannotated nets.
3. PrimePower accesses the Synopsys library power tables and determines the power consumption for leaf-level cells which are summed to obtain the total design power. When performing event-based analysis, PrimePower processes every activity in the activity file to build an accurate power profile over time and to determine the peak power consumption.

### 2.2.2 Power analysis flow

PrimePower has two methods for power analysis:

- Event-based power analysis: PrimePower uses an event driven simulation algorithm to calculate the power consumption for each event. It provides accurate and detailed time-based power information. Event-based analysis is vector dependent and can produce an average and peak power report.

- Statistical activity based power analysis: PrimePower uses a statistic and probability estimation algorithm to calculate the average power and construct the average cycle power waveform. It is vector free (no vector input) or vector weak dependent.

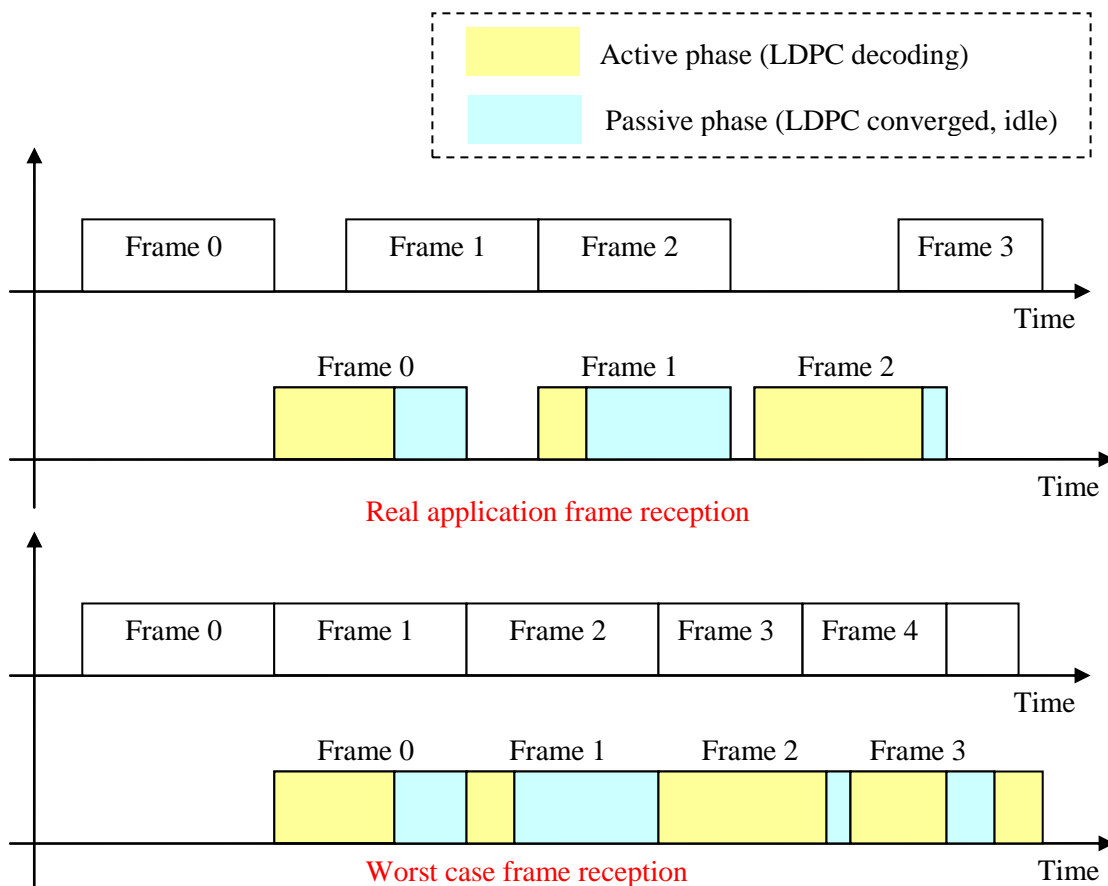
Within DaVinci, an event-based power analysis will be carried out, allowing accurate power information based on sensible RTL simulations.

PrimePower calculates static power and dynamic power based on the library information, netlist, design constraints and activity. The following steps are performed during event-based analysis:

1. Captures events from the simulation as VCD data.
2. Sets up the ramp times of the input ports.
3. Annotates the wire capacitance.
4. Propagates the ramp times.
5. Matches the internal power table with the simulation events.
6. Calculates the switching power.

### 2.2.3 DaVinci scenarios

Power estimation on a gate netlist requires low-level simulations which could take long run times. For this reason, we have selected two modes for which the power estimation will be done. These are **mode 9** and **mode 15**, respectively **rate 2/3, N=288** symbols and **rate 3/4, N=384** symbols.



**Figure 3 : Frames reception, decoding active and passive phases**

Power consumption depends on lots of parameters, most importantly: temperature, voltage and SNR conditions. This last parameter is very important in DaVinci as low-noise communication means fewer decoding iterations required, thus lower active decoding time. In order to get sensible power consumption figures, we will distinguish the power in active and passive decoding time. The active phase corresponds to the time when the LDPC decoder is decoding and running iterations. Passive phase corresponds to idle period of the frame decoding, i.e. the current frame has been decoded, no iteration running, but data are still transferred to the next block while a new frame is being stored in the input RAMs of the decoder. In Figure 3 we have represented two scenarios: the real application in which “holes” are present between frames, and the worst case frames reception, that is, a continuous data stream. Note that the active period is variable from one frame to another (for the same SNR) and is highly dependent on SNR conditions.

The power estimation described in this report will be performed with the worst case assumption, as we have not enough visibility on the “holes” duration of a real case. However, a scaling factor can be applied to the power figures, in order to take into account these idle periods.

Power figures will be given as a mean power by iteration for the active periods and passive periods. Then, knowing the mean number of iterations (for a given mode and SNR), we can deduce the mean power during a continuous data stream by applying the formula:

$$Frame\_power = \frac{active\_power * mean\_nb\_iter + passive\_power * (18 - mean\_nb\_iter)}{18}$$

We assume that for a continuous data stream at the maximum throughput, the frame reception duration corresponds to 18 iterations decoding time, which is the maximum specified to support 100 Mbit/s user data throughput.

## 2.3 Power estimation results

### 2.3.1 Gate netlist properties

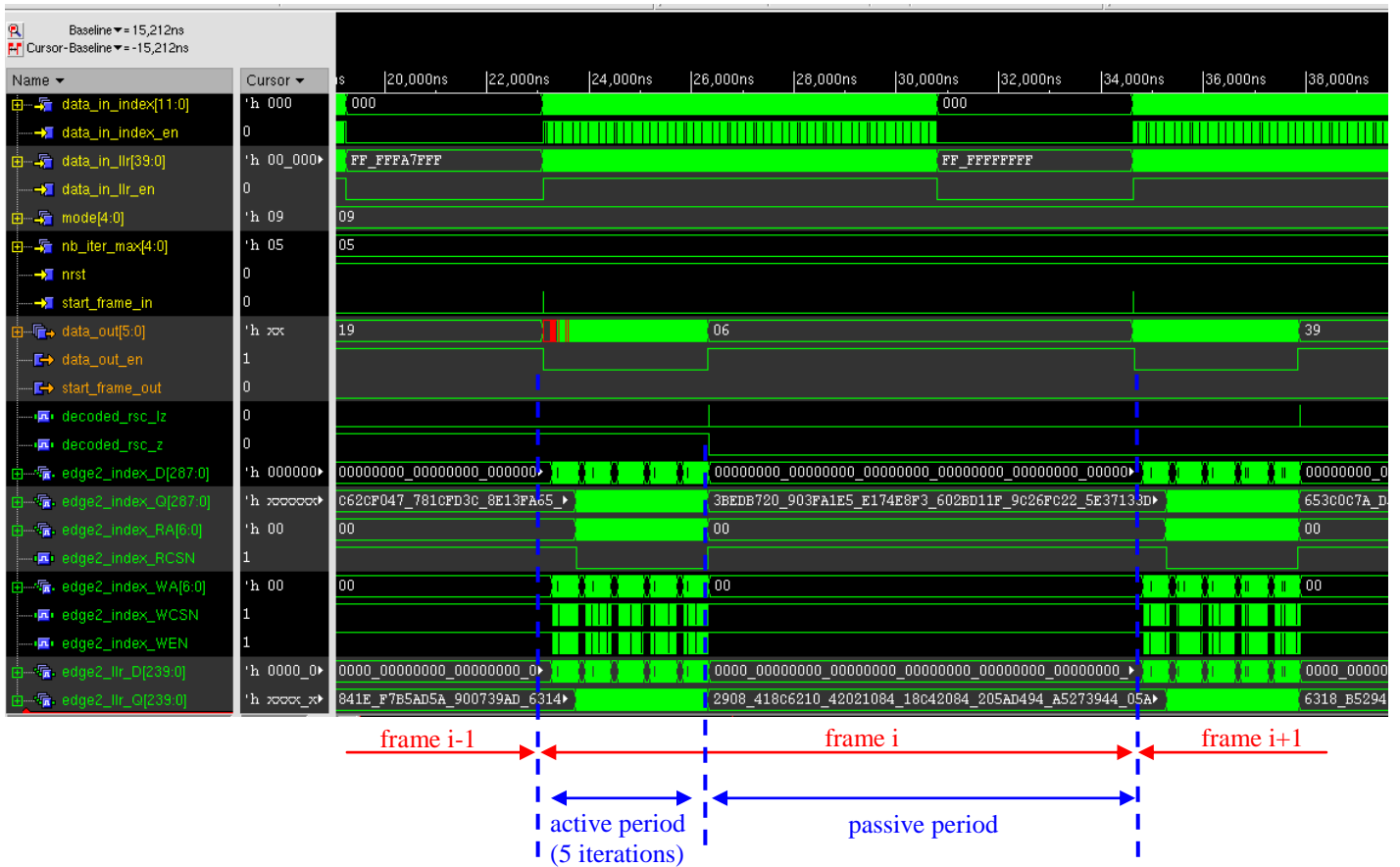
We recall here the characteristics of the gate netlist used for the power estimation, which has been obtained during task 3.1.1 and described in deliverable D3.1.1.

- Clock frequency: 300 MHz. The logic synthesis has been done in worst case timing conditions, i.e. at 125C under 1.1V. Thus real circuit may support higher frequency in nominal conditions.
- User throughput (decoded data): 100 Mbit/s in all modes
- Post-synthesis area: 1.28 mm<sup>2</sup> on a 45nm LP technology

The power estimation will be done using nominal conditions model, as it is the most probable case in the real application. Consequently, we assume a 1.1 V voltage supply at 25 °C.

### 2.3.2 Measured power in active and passive periods

The following figure illustrates the kind of simulations required at RTL level for the power estimation. We can easily distinguish on the chronograms the active and passive periods, as the internal signals are active (changing state) or stable. We can also identify the start of decoding iterations as some idle cycles are present between each of them.



**Figure 4 : RTL simulation showing active and passive periods**

The power in the active period has been measured over 5 iterations for both modes. Table 1 contains the results of the estimation in active period, detailing static and dynamic power.

	Dynamic power	Static power	Total power
Mode 9	0.216	1.64E-4	0.216
Mode 15	0.202	1.65E-4	0.202

**Table 1: Power measured in the active period, in Watt**

The power in the passive period has been measured during a time equivalent to 5 iterations. Table 2 contains the results of the estimation in passive period, detailing static and dynamic power.

	Dynamic power	Static power	Total power
Mode 9	0.0669	1.60E-4	0.067
Mode 15	0.0658	1.60E-4	0.066

**Table 2: Power measured in the passive period, in Watt**

**Comments:**

- From these results, we deduce that the power consumption in mode 9 and mode 15 is nearly the same: around **0.2 W** in the active period of the frame and **0.067 W** in the passive period. This result was predictable as, in both modes, all processors instantiated are running about the same percentage of time. This conclusion can be extended to all modes.
- The static power, in all cases, is negligible compared to the dynamic power. However, the value indicated is interesting in case of long inactivity periods of the LDPC are identified in the application.
- In the active period, we measured 1.66E-2 W for a macro CN processor and 1E-2 W for a macro VN processor. The DaVinci decoder contains 4 processors of each type, which results in a power of 0.106 W for the 8 processors. We note that more than half of the total power consumption of the LDPC decoder is due to the processors.

**2.3.3 Mean number of iterations study**

C simulations have been run to determine the mean number of iterations required to decode a frame for several modes and in several SNR conditions. From these values and the formula given in 2.2.3, we could deduce the mean power over a frame decoding, assuming an input throughput of 100 Mbit/s (user throughput) and continuous data stream. The power value obtained is then realistic and, as expected, closely related to the SNR value.

We also determined the energy, in Joules, required to decode a frame by multiplying the mean power by the decoding frame duration.

The studies in mode 9 and mode 15 are summarized in Table 3 and Table 4. The mean number of iterations decreases with the SNR and, consequently, the power consumption decreases too. For SNRs in the waterfall region, we get power figures below 0.1 W. The energy/frame is indicative and can be used to compare DaVinci decoder with other LDPC decoders in the literature.

We underline the small mean number of iterations compared to the 18 available iterations to decode a frame.

<b>Eb/N0 (dB)</b>	2	2.2	2.4	2.6	2.8	3
<b>FER</b>	1.5E-1	3.3E-2	4.5E-3	3.4E-4	< 1E-5	< 1E-5
<b>Mean n.o. iterations</b>	5.87	5.36	4.47	3.81	3.37	3.06
<b>Mean frame power (W)</b>	0.1156	0.1114	0.104	0.0985	0.0949	0.0923
<b>Energy/frame (μJ)</b>	1.33	1.28	1.20	1.11	1.09	1.06

**Table 3 : Simulation results and power consumption, mode 9, K=192**

<b>Eb/N0 (dB)</b>	2.4	2.6	2.8	3	3.2	3.4
<b>FER</b>	2.5E-1	5.2E-2	5.7E-3	3.8E-4	< 1E-5	< 1E-5
<b>Mean n.o. iterations</b>	5.42	5.3	4.35	3.64	3.2	2.91
<b>Mean frame power (W)</b>	0.107	0.106	0.0989	0.0935	0.0902	0.088
<b>Energy/frame (<math>\mu</math>J)</b>	1.85	1.83	1.71	1.62	1.56	1.52

**Table 4 : Simulation results and power consumption, mode 15, K=288**

Table 5 shows the results of the study for a fixed Eb/N0, i.e. 2.2 dB, but for different modes ranging from mode1 to mode4. Actually, we wanted to see the evolution of the power with the frame length, but for a constant code-rate (1/2 here). We assumed a power of 0.22 W in the active period and 0.067 W in the passive period.

As we can see, the mean number of iterations and the mean power increase slightly with the code length, but the power remains around 0.1 W. It is known that longer LDPC frames require more iterations to converge. In comparison, the energy increases a lot with the frame length, as the decoding time increases proportionally.

<b>Mode</b>	Mode 1 N=96	Mode 2 N=192	Mode 3 N=288	Mode 4 N=384
<b>FER</b>	3.3E-4	1.3E-4	< 1E-5	< 1E-5
<b>Mean n.o. iterations</b>	3.72	4.03	4.23	4.38
<b>Mean frame power (W)</b>	0.0986	0.101	0.103	0.104
<b>Consumption (<math>\mu</math>J)</b>	0.284	0.583	0.89	1.2

**Table 5 : Simulation results and power consumption, Eb/N0=2.2dB**

### 3. Conclusion

In this report, we explained the methodology used to evaluate the power consumption of the DaVinci decoder in its ASIC version. The study has been carried out on the optimized gate netlist obtained in sub-task 3.1.1. Assuming realistic SNR conditions with a continuous user data stream at 100 Mbit/s, we obtained around 100 mW total power for the LDPC decoder, which is in the range of the expected power values.

When the gate netlist has been created, we focused on the silicon area optimization, without taking care of the power. We have not applied particular low-power design techniques; nevertheless the resulting power is clearly acceptable. If the target of such LDPC decoder is a battery-powered device, an area of improvement could be the power consumption at RTL and gate level.