Improving the energy–consumption of a passive RFID–tag

Michael Klamminger

Graz University of Technology

Institute of Electronics

Supervisor: Ass.Prof. Dipl. Ing. Dr.techn. Peter Söser

Graz, March 2013

Abstract

Radio Frequency Identification (RFID) is increasingly widespread in various applications. For many of these applications, a wide range is essential. Thus this range can also be made possible without an active energy supply, may this RFID tag always require less power. A major consumer of energy in the Integrated Circuits (ICs) is the digital section. The more efficiently it works, the more efficiently the whole system works.

The Department of Contactless and RF exploration (CRE) at Infineon Technologies Austria is investigating the possibilities of RFID. A system built on the EPC protocol serves as a basis for this research. The digital part of this system already existed. To evaluate other possibilities of RFID the digital part needs to be optimized during work.

Kurzfassung

Radio Frequency Identification (RFID) findet immer mehr Verbreitung in verschiedensten Anwendungen. Für viele dieser Anwendungen ist eine große Reichweite entscheidend. Damit diese Reichweite auch ohne eine aktive Energieversorgung ermöglicht werden kann, dürfen diese RFID-Tags immer weniger Leistung benötigen. Ein großer Energiekonsument in den ICs ist der Digitalteil. Je effizienter er funktioniert, desto effizienter kann auch das ganze System arbeiten.

Die Abteilung Contactless and RF exploration (CRE) bei Infineon Technologies Austria forscht an den Möglichkeiten von RFID. Ein auf dem EPC-Protokoll aufbauendes System dient als Basis für diese Forschung. Dafür wurde bereits ein Digitalteil entwickelt der dieses Protokoll implementiert. Um weitere Möglichkeiten von RFID zu evaluieren soll dieser Digitalteil im Rahmen dieser Arbeit optimiert werden.

Table of Contents

List of Figures

Acronyms

Notation	Description
ASK	Amplitude Shift keying.
CG	Clock Gate.
CGC	Clock Gating Cell.
CRC	Cyclic Redundancy Check.
CRE	Contactless and RF exploration.
CTS	Comprehensive Transponder System.
DR	Divide ratio.
EPC	Electronic Product Code™.
HF	High Frequency.
IC	Integrated Circuit.
LF	Link Frequency.
LFSR	Linear Feedback Shift Register.
NVM	Non-Volatile Memory.
PFK	Pulse Frequency Keying.
RF	Radio Frequency.
RFID	Radio Frequency Identification.
RN	Random Number.
RNG	Random Number Generator.
R⇒T	RFID Reader to tag.
T⇒R	Tag to RFID Reader.
TUG	Graz University of Technology.

Notation	**Description**
UHF	Ultra High Frequency.
VHDL	Very High Speed Integrated Circuit Hardware Description Language.

Chapter 1

Introduction

1.1 Motivation — The need for the reduction of the power of a passive Radio Frequency Identification (RFID) chip

In the Comprehensive Transponder System (CTS) project from Graz University of Technology (TUG) and the Contactless and RF exploration (CRE) department at Infineon Technologies Austria an RFID–tag was developed. It implies the Electronic Product Code™ (EPC) HF[6] and UHF[5] standards. The tag is passive, so it gets its energy from the reader. If an HF field is present the tag is in the HF mode. In this case the tag gets its power through inductive coupling. For replaying it uses load modulation. Here the main problem is not to get enough power to the tag, but decode the reply in the reader.

The UHF mode is the default mode of the chip. It is always in this mode when a field is present and its not detected as an HF field. Here the tag gets its power via the H–field and replies with back scattering, which allows greater distances. The H–field induces a voltage in the antenna. A voltage converter rectifies it and limits the output voltage for the digital circuit (Figure 1.2). The voltage converter has an efficiency of approximately 20 %, so for every µW consumed from the digital part, 5 µW at the antenna are needed.

Figure 1.2 shows an Amplitude Shift keying (ASK) modulated signal. One can see, that while the tag is receiving data there is even less power. The same is even more true for the transmission. A zero is realized through a short circuiting of the antenna. When the shunt closes, no further energy will be injected.

To get the system through this phases, it is necessary, that there is some energy storage. Here it is a capacitor, as on chip batteries are not available jet. But it is hard to realize big capacities on a chip. A voltage detector monitors the voltage of the capacitor and resets the chip, if a certain voltage level is undershot.

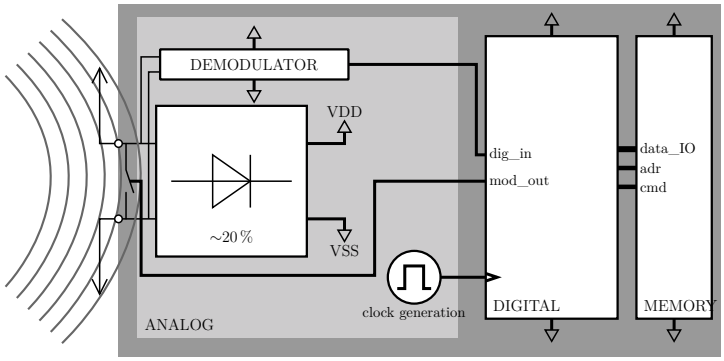

Figure 1.1: Block-diagram of the tag

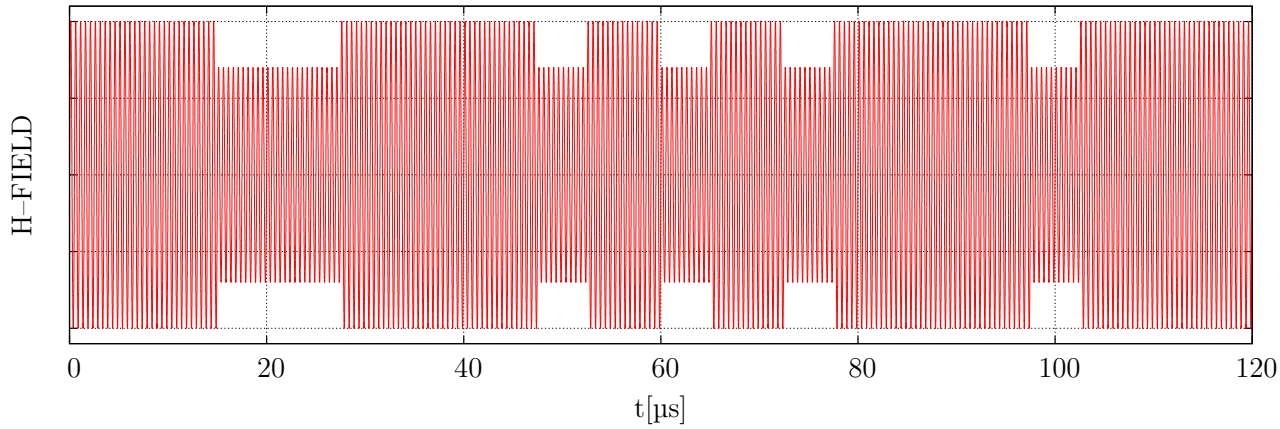

Figure 1.2: Sample of an ASK modulated signal with 2 MHz and a modulation index of 20 %

So there is the need for lowering the over all power consumption, to extend the working range. A flattening of the energy consumption during communication also helps that the tag doesn't reset itself.

1.2 Short introduction to the standards used in this project

The EPC standards we are using in this project are defined by EPCglobal®. The standards can also be found on their homepage. The EPC HF and UHF are two different standards, but they are fundamentally compatible.

Both use an ASK modulated Pulse Interval Encoding signal for the R⇒T communication.

The data rates of the communication highly depends on the payload. Three basic periods are given by 6.25 µs, 12.5 µs and 25 µs for the EPC UHF standard. This is the duration of a zero symbol (T_{zero}). For the EPC HF standard T_{zero} can be from 8 µs to 25 µs. The time of a one symbol (T_{one}) is in both cases 1.5 to 2 times T_{zero}. The length of T_{zero} and T_{one} are synchronised with the R⇒T preamble.

The R⇒T preamble can also contain a time T_{TRcal}. The EPC UHF standard defines the T⇒R Link Frequency (LF) as:

$$LF = \frac{DR}{T_{TRcal}} \tag{1.2.1}$$

The Divide ratio (DR) is sent from the RFID Reader to the tag with the payload. In HF mode there are only two possible LFs, 424 kHz and 847 kHz. DR is only used to select the LF and T_{TRcal} is ignored.

For the base band encoding of the T⇒R data either FM0, Miller for both, or Manchester in HF are used. Miller Code and Manchester Code use two, four, or eight subcarriers per symbol. That's why the the data rates are lower than the LF.

The link timing between tag and reader has also to work within given parameters. When a reader sends a command to start an inventory round the tag has a time T_1 to answer the reader. Then the reader has a time T_2 to send its next command.

$$T_{pri} = \frac{1}{LF} \tag{1.2.2a}$$

$$T_1 = 10\ T_{pri} \tag{1.2.2b}$$

$$T_2 = [3\ T_{pri}; 20\ T_{pri}] \tag{1.2.2c}$$

The EPC HF and UHF standards describe eight different states in which a tag can be. Depending on the state the tags are in, they react different to the R⇒T commands. In the EPC standards 14 commands in three groups are defined. The first group is to select a population of tags, the second is to inventory one tag and the third is to access the tag. With the access commands, the memory is read or written. A new Random Number (RN) can be requested. The tags accessibility can be limited, or completely prohibited, for future use.

The EPC protocols are quite complex for an RFID protocol. This is because of its good anti collision capabilities.

For the CTS project [5] and a draft version of [6] were used.

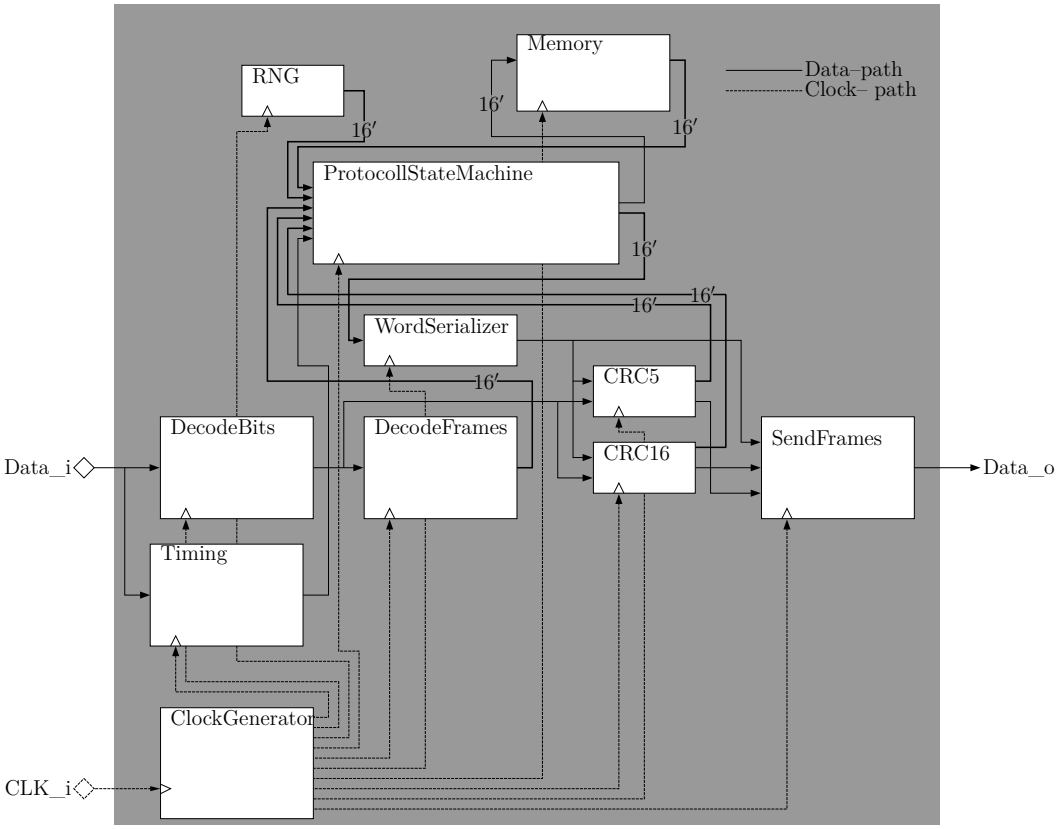

Figure 1.3: Block-diagram of the existing design

1.3 Introduction to the existing CTS digital design

As the work is about an optimization of a digital part, has to be already an existing design. It was conceptualized by Johann Heyszl in [4] and implemented by Thomas Andrejka in [1].

An overview of the modules implemented in the project is shown in Figure 1.3. The functions of each block are as follows:

- **ProtocolStateMachine** is the main controller of the design. It does the protocol decoding, the state control and the reply preparation. All the controlling is done by a single, large state machine. To control everything it always has to be clocked, but with a low frequency of $clk/14$ when in HF and $clk/17$ when in UHF mode.
- **Timing** is the unit to measure the timings between R⇒T and T⇒R communication. It also is capable of the detection of timing violations for the given mode. The main component of the module is a counter and a compare logic. As it only measures the time between the communications, it is always disabled during the communication.

- **ClockGenerator** divides the main clock and provides it to the other modules. The input frequency of the design is 1.659 MHz, recovered from the 13.56 MHz carrier, in HF or about 1.92 MHz, generated through an oscillator, in UHF. It suppresses the output of the clocks for individual units, if not needed according to the ProtocolStateMachine.
- **DecodeBits** receive the Pulse Frequency Keying (PFK) modulated, binary input stream. The main component of these unit is the counter, which counts the high and low time of the input and generates a bit stream of it. The time between communications is also measured and stored in here. This is necessary for the synchronization between reader and tag. For the decoding it is sufficient to clock the unit with $clk/2$, but it need not only be active during receiving data, but also when waiting for a new input stream.
- **DecodeFrames** get the binary coded data and a handshake from DecodeBits, to determine when new data arrive. It analyze the data as they arrive to find out some parameter of the communication, such as the received command. The captured parameters and the remaining parallelized data are passed to the ProtocolStateMachine for further processing. This needs also a handshake signal between DecodeFrames and the ProtocolStateMachine. The clock is only applied when receiving and depends on the incoming data.
- **CRC5** and **CRC16** are the modules to calculate the Cyclic Redundancy Check (CRC). They are needed to verify the incoming data and to generate a checksum for the outgoing data. Depending on the command, it can be protected via a CRC16, a CRC5, or not at all. When receiving they get the same serial data as DecodeFrames and also the same clock. The output of these modules is an unverified checksum. When transmitting they get a serial stream from WorldSerializer and a clock to be fast enough for the transmission rate.
- **WordSerializer** has a 16 bit parallel data input. Internal it consists of two 16 bit registers that are loaded and shifted alternate. The data from the shift register are outputted serial. While sending, the clock speed is the same as for the CRC unit. Otherwise it is disabled.
- **SendFrames** receive the data to transmit via the WordSerializer, or via one of the CRC modules, in serial and generate an output stream that is modulated in one of FM0, Miller2, Miller4, Miller8, Manchester4 or Manchester8 depending on the settings.
- **RNG** continuously generates pseudo random numbers at a speed of $clk/8$.
- **Memory** either controls the Non-Volatile Memory (NVM), in case one is attached, or provides pseudo data by itself. It gets the same clock as the ProtocolStateMachine.

1.4 Requirements for the work

The work is about analyzing the implementation of the existing design. Founded potentials to optimize the design have to be verified and implemented if they have the potential to pay off. Of course the full functionality must be preserved.

Chapter 2

Analyzing and Improvements of the Digital Design

Before it is possible to reduce the power, it is necessary to analyze where the power gets lost. Most of the power consumed by a digital circuit is transformed to thermal energy.

$$P = P_{dyn} + P_{stat} \tag{2.0.1a}$$
$$P_{dyn} \approx \alpha \; 1/2 \; (C_G + C_N) \; VDD^2 \; f_{clk} \tag{2.0.1b}$$
$$P_{stat} = I_{leak} \; VDD \tag{2.0.1c}$$

Equation 2.0.1 shows a rough estimation of how the power is spend. P_{dyn} is the dynamic power and P_{stat} is the static power. C_G is the sum of all gate capacities and C_N are the parasitic capacities of the wires. I_{leak} is the total of the leaking current in the circuit. α is a proportionality of the switching activities and is somewhere between 0 (no switching at all) and 1 (switching at full clock rate).

C_G and I_{leak} are proportional to the number of gates in the design and C_N to the length of the wires.

2.1 Voltage

VDD has the biggest influence to the power consumption. It also influences the maximum possible clock speed, as for higher switching rates, higher voltages are needed. But in our case VDD is already at the lowest possible value for the used technology.

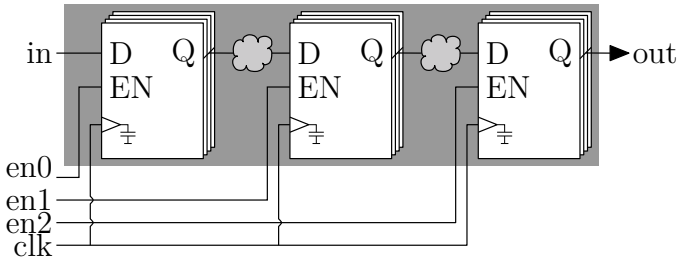

Figure 2.1: Scheme of multiple buffers without clock gating

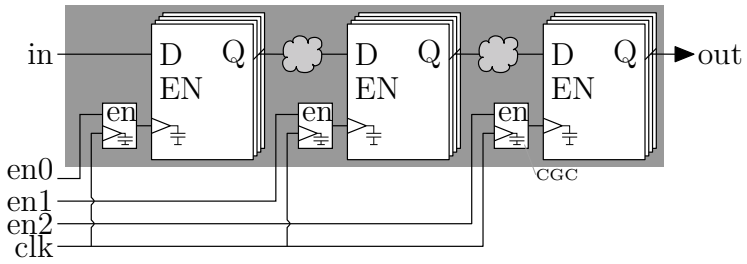

Figure 2.2: Scheme of multiple buffers with clock gating

2.2 Clock

The influence of the clock frequency (f_{clk}) directs proportional to the dynamic power consumption. But it was exactly calculated to achieve all the requirements of the Electronic Product Code™ (EPC) standards.

Even though it is not possible to reduce the input clock speed of the system, it is possible to reduce the clock edges which reach the gates. The consisting design uses a clock–generation–unit. This unit divides the clock into several sub-clocks, which are delivered to the other units. With this approach there are several clock domains on the chip. To synchronize between the domains, additional handshake signals are necessary. For the synthesis also automated clock gating was applied. A potentially better approach might be to use the handshake lines for automatic clock gating.

Figure 2.2 shows what the circuit in Figure 2.1 looks like, when the clock signal is gated. Clock Gates (CGs) are additional inserted to the design. But they don't influence the size of the design much, as it would be necessary to insert clock drivers anyway.

To ensure that the synthesis tools are able to find parts which can be gated, it is important to follow a certain coding style.

Traditionally a register in VHDL would be described like this:

```
if rising_edge(clk_i) then
  counter <= counter_next;
end if;

counter <= counter + 1 when counter < 10
           else 0;

en <= '1' when counter = 0
      else '0';
```

Listing 2.1: *EN* generation

```
if rising_edge(clk_i) then
  reg <= data_i;
end if;

if en = '1' then
  next_reg <= data_i;
else
  next_reg <= reg;
end if;
```

Listing 2.2: Not gated clock

In this code sample the incoming data would always be updated when *en* is high. *en* is only every 10^{th} clock circle high. But the clock still would be delivered to the register every circle, as the synthesis tool lacks the information to insert a CG at this point.

If you change the code style as follows, CG will be enabled.

```
if rising_edge(clk_i) then
  counter <= counter_next;
end if;

counter <= counter + 1 when counter < 10
           else 0;

en <= '1' when counter = 0
      else '0';
```

Listing 2.3: *EN* generation

```
if rising_edge(clk_i) then
  if en = '1' then
    reg <= reg_next;
  end if;
end if;

reg_next <= data_i;
```

Listing 2.4: VHDL-code prepared for clock gating

In the sequential part, it is important, that there is an 'if' condition without an 'else'. So the synthesis tool has the information that the register only changes when *en* is high and no clock need to be applied otherwise.

By changing the code like this, a better CG coverage is achieved.

Additional to the change of the code the manual inserted CGs can be removed. The static divided clocks can be served as enabling signals to the units and the conditional clocks are simply replaced by stirring signals between the units.

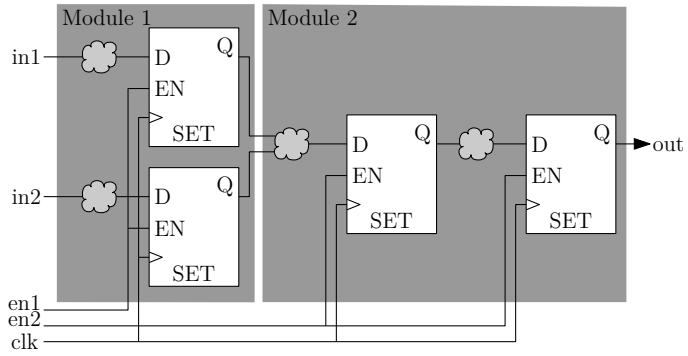

Figure 2.3: Scheme of pipeline with multiple buffers

2.3 Capacity and Leakage

The reduction of cells will decrease C_G and I_{leak}, but will increase α and vice-versa. Fast systems often need additional flip-flops to lower the utilization of each of them. Because of slower switching rates the voltage can be reduced. However for this design it would be more efficient to reduce some cells with low utilization, since already at a switching interval less than 20 kHz the static power would outweigh.

A reduction of the gates would also decrease C_N, because there are less wires needed to connect the gates.

Section 2.4.1 shows which modules of the system are not optimal and should be replaced or rather could be combined.

2.4 Optimization of the Design Modules

2.4.1 Tracking redundant cells to remove

In a low power and low speed design it is not necessary to buffer every value after view gates. Figure 2.3 shows how a solution for a rather fast design would look like. It is necessary that there are not too much combinatorial logic between two cells. If it were, the signal would not arrive at the input of the next flip-flop in time.

A low speed design does not need to buffer that much, because the time that a signal has to pass through the combinatorial logic is much longer.

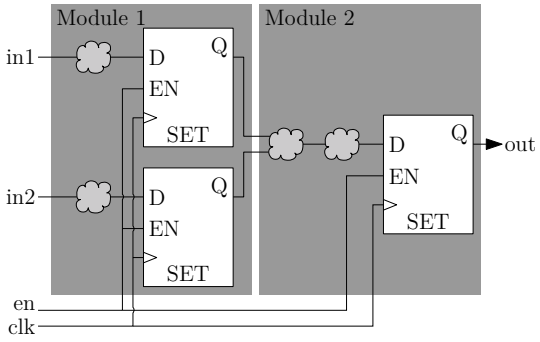

Figure 2.4: Low speed design with more logic between two buffers

For example in the DecodeBits unit the time for the calculation of the Link Frequency (LF) was stored. The LF is calculated with:

$$LF = \frac{DR}{T_{TRcal}} \tag{2.4.1}$$

DR is the divide ratio, which also only changes with the same reader command. Because of that the LF can only change there and is static during the rest of the time.

The LF is only needed to send data. That happens at a different time. That made it possible to remove the register that stored the value of LF.

2.4.2 Modules Timing and DecodeBits

The incoming serial communication is processed by two modules. The Timing unit measures the time between two signals and DecodeBits which demodulate the Pulse Interval Encoding signal and output the binary signal. For both tasks a counter is needed, but never at the same time. It is not possible to deactivate the DecodeBits unit during the pause between two communications, as it need to be in standby for the next incoming data. The combination of these two units would save about 16 flip-flops.

2.4.3 Modules CRC5, CRC16 and RNG

The communication between reader and tag is protected in both directions by a Cyclic Redundancy Check (CRC). Some commands use a CRC5 and others a CRC16. Figure 2.5 shows the scheme of a CRC5 calculation. One can see, that the calculation is basically just a shift-register with some additional XOR gates. The CRC16 works the same. The

places where the gates are put, are determined by a so called polynomial. Equation 2.4.2 shows how to calculate a CRC16.

$$ACCU[0:15] := [(ACCU[15] \oplus data); ACCU[0:14]] \oplus POLY[0:15] \tag{2.4.2}$$

If the polynomial is configurable and if it is possible to only activate the lowest five bit, it is possible to calculate five or 16 bit wide CRCs. An other problem is, at the beginning of the message you can not know which instruction follows. But all command codes starting with 01 will be protected through a CRC5. So it is possible to start with a CRC16 check and if the first two bit of the communication are 01, reset the CRC unit to the value that the CRC5 would have if 01 was checked.

$$ACCU[0:15] := [ACCU[15]; ACCU[0:14]] \oplus POLY[0:15] \tag{2.4.3}$$

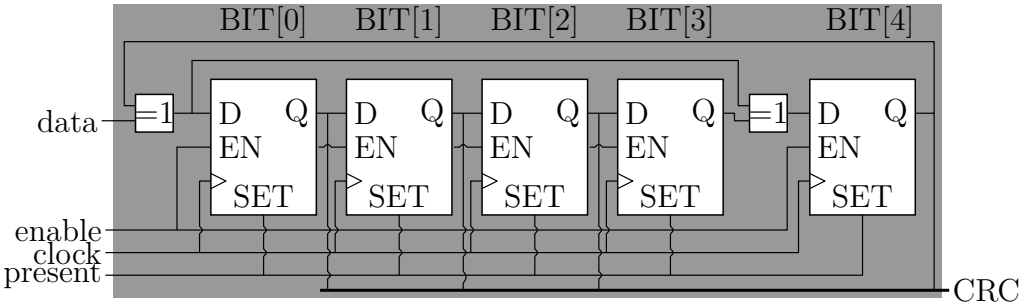

Figure 2.5: Scheme of CRC5 calculation

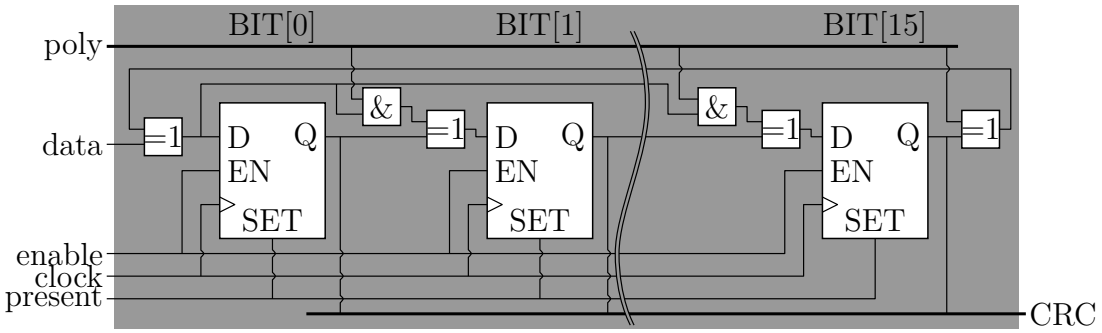

Figure 2.6: Scheme of configurable CRC and LFSR generator

Linear Feedback Shift Registers (LFSRs) are a common way to calculate pseudo random numbers. A LFSR can be calculated as in equation 2.4.3. If 2.4.2 and 2.4.3 are compared, one can see that there is almost no difference between them. As the Random Number Generator (RNG) is never active during communication, it is also possible to combine it with the CRC module.

A scheme of the combined logic is shown in Figure 2.6.

Chapter 3

Results and Conclusion

The resulting design, as shown in Figure 3.1, was never produced. At the time my work was finished, the main focus of the team has changed towards RFID with sensor capabilities.

The optimizations were integrated in further chips. But this chips have so much other parts on it, that it is impossible to tell anything about the outcome of my work. Therefore there are no measurement results, or even estimations according to the reduction of the power consumption of the system.

With the design changes in the clock tree, a clock gate utilization of above 95 % could be achieved. Here the most energy is saved.

With the reduction of redundant cells only about fifty flip-flops were removed. With about 500 flip-flops in the original design, that are 10 %. But when this cells were removed, some additional combinatorial logic was added. So the exact reduction of power could not be told.

Real power saving can only be done during the design phase. The biggest part in the design is the ProtocollStateMachine. There is a huge potential to optimize its energy efficiency. But due to a not modular concept and a nested implementation of the state machines for the initial states and the command decoding, it was impossible to change anything without rewriting the whole module new.

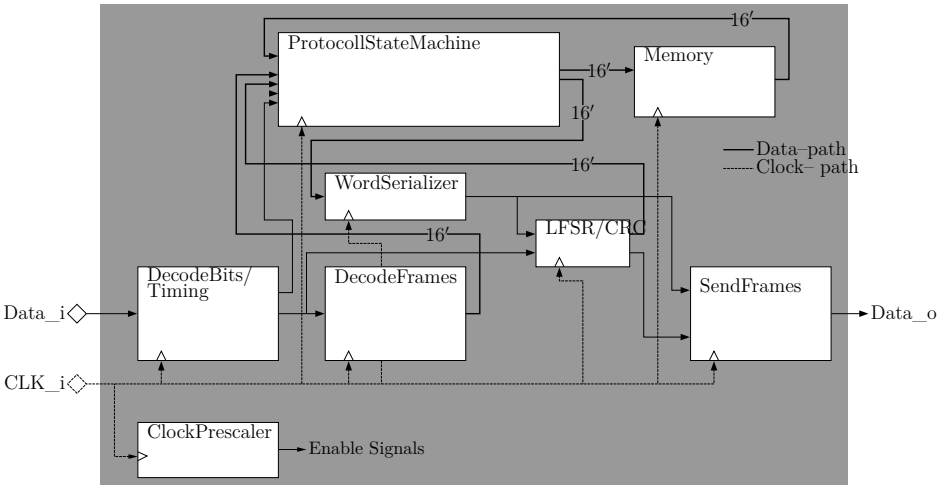

Figure 3.1: New top level of the CTS digital part

Glossary

Notation	Description
EPCglobal®	"is leading the development of industry-driven standards for the Electronic Product Code™(EPC) to support the use of Radio Frequency Identification (RFID) in today's fast-moving, information rich, trading networks." [2].
FM0 code	"A binary 0 is coded by a transition of either type in the half bit period, a binary 1 is coded by the lack of a transition. Furthermore, the level is inverted at the start of every bit period, so that the bit pulse can be more easily reconstructed in the receiver" [3].
inventory round	is a sequence of several commands. It is used to select one tag out of a group of tags. Before the reader can apply any actions to a tag, an inventory is required.
link timing	is the time the participants in the communication have to deliver a valid reply. In case the link time is not valid the communication has to reset and start again.
Manchester Code	"is a line code in which the encoding of each data bit has at least one transition and occupies the same time. It therefore has no DC component, and is self-clocking, which means that it may be inductive or capacitive coupled, and that a clock signal can be recovered from the encoded data." [8].

Notation	**Description**
Miller Code	"is the encoding of binary data to form a two-level signal where a "0" causes no change of signal level unless it is followed by another "0" in which case a transition to the other level takes place at the end of the first bit period; and a "1" causes a transition from one level to the other in the middle of the bit period." [7].
Pulse Interval Encoding	is the modulation technique used for RFID Reader to tag communication. Only the length of the high amplitude of the signal is deciding whether it is a one or a zero symbol.
RFID Reader	"A reader typically contains a RF module (transmitter and receiver), a control unit and a coupling element to the transponder." [3].
state machine	is a finite automaton which is able to process a certain problem. In hardware design often state machines are used, as they are usual more efficient than programmable processors.
tag	is the passive unit in a Radio Frequency Identification (RFID) environment. It never initialises the communication. A tag can either be active powered via a battery, or via the field the RFID Reader produces.

Bibliography

[1] Andrejka, T. „Implementierung der epc class-1 generation-2 rfid für ein comprehensive transponder system.“ MA thesis. Fachhochschule Hagenberg, 2007.

[2] *Epcglobal / products & solutions / gs1 - the global language of business*. 2013. http://www.gs1.org/epcglobal (visited on 02/12/2013).

[3] Finkenzeller, K. *Rfid- handbuch*. City: Hanser Fachbuchverlag, 2002.

[4] Heyszl, J. „Linksystem research and rtl design of a combined passive hf - uhf rfid tag.“ MA thesis. Graz University of Technology, 2007.

[5] Inc., E. *Epc™ radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz – 960 mhz*. EPCglobal Inc™, 2004.

[6] Inc., E. *Epc™ radio-frequency identity protocols epc class-1 hf rfid air interface protocol for communications at 13.56 mhz*. EPCglobal Inc™, 2011.

[7] Wikipedia. *Delay encoding — wikipedia, the free encyclopedia*. [Online; accessed 24-March-2013]. 2013. http://en.wikipedia.org/w/index.php?title=Delay_encoding&oldid=540897313.

[8] Wikipedia. *Manchester code — wikipedia, the free encyclopedia*. [Online; accessed 24-March-2013]. 2013. http://en.wikipedia.org/w/index.php?title=Manchester_code&oldid=54299000