

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

BP30 SIM Driver Specification

Edition 2006

Published by Neonseven s.r.l.,
Viale Stazione di Prosecco, 15
34010 Sgonico (Trieste) Italy

© Neonseven.
All Rights Reserved.

For questions on technology, delivery and prices please contact the Neonseven Offices in Italy Sgonico and Gorizia

Attention Please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact Neonseven.

Neonseven technologies may only be used in life-support devices or systems with the express written approval of Neonseven, if a failure of such technologies can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Author	Enrico Bandera	Department:	S2	Page: 1/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –		Confidential	

Table of Contents

1	Document Mission/Scope	3
1.1	Mission	3
1.2	Scope	3
2	List of Acronyms	3
3	Introduction	3
4	Architecture	3
4.1	The EGoldRadio SIM interface	3
4.1.1	SIM Character Mode	4
4.1.2	SIM T=0 Protocol Mode	5
4.2	Card sessions phases	5
4.3	Software structure	6
4.4	Main controller	7
4.4.1	Activation	7
4.4.2	PPS negotiation	8
4.4.3	APDU command execution	8
4.4.4	Deactivation	8
4.4.5	Main control State machine	8
4.5	SIM access	9
4.5.1	ATR handler	9
4.5.2	PPS handler	12
4.5.3	T=0 instruction handler	15
4.6	HW drivers	17
4.6.1	UART driver	17
4.6.2	Timer driver	17
5	Interface	18
5.1	SI interface	18
5.1.1	SI Synchronous interface	18
5.1.2	SI Asynchronous interface	19
5.2	ATC interface	20
5.3	Power down interface	20
6	References	22
6.1	External	22
6.2	Internal	22
7	Document change report	22
8	Approval	22

1 Document Mission/Scope

1.1 Mission

This document contains the specification of the SIM driver module, which provides the entry point for accessing the SIM Card.

1.2 Scope

This document is addressed to SW developers who want to either interface to or upgrade the SIM Card module; therefore both the interface and the implementation of the SIM hardware driver are described in details.

2 List of Acronyms

Abbreviation / Term	Explanation / Definition
APDU	Application Protocol Data Unit
ATR	Answer To Reset
CI	Card Interface
PEC	Peripheral Event Controller
PPS	Protocol and Parameter Selection
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver Transmitter

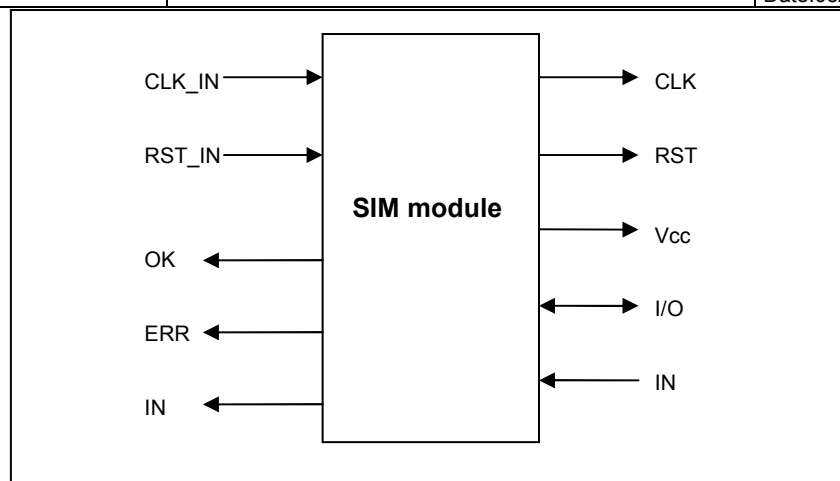
3 Introduction

The SIM hardware driver module basically handles the data exchange with the SIM Card by programming and controlling the half-duplex UART, which constitutes the actual physical interface towards the SIM, according to the T=0 Protocol. It is responsibility of the Client to properly format the SIM Commands, which are transferred transparently to the SIM; the driver takes care of the low level ATR, PPS and T=0 Protocol handling, such as byte acknowledgement, parity errors, character and signals timing, clock stop mode and so on. For a description of the T=0 Communication Protocol please refer to ISO-IEC 7816-3, for the FileSystem to GSM 11.11, for Voltage issues to GSM 11.12 and GSM 11.18.

4 Architecture

4.1 The EGoldRadio SIM interface

The EGoldRadio includes a SIM module shown in Figure 4-1; Clock, Voltage, I/O and Reset signals are provided and there are 3 interrupt sources: OK for correct transmission/reception, ERR for parity/overflow errors and T=0 end, IN for SIM presence detection.


Figure 4-1

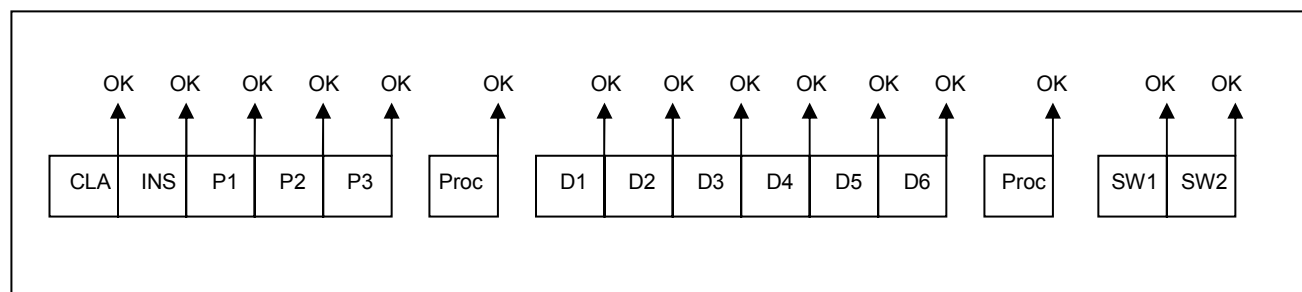
The SIM Card is fed with a 3.25 MHz clock, derived from a 13 MHz clock. The following features are provided by the EGoldRadio SIM module:

- 1) Automatic parity error detection and error signalling in RX mode.
- 2) Automatic character repetition on parity errors in TX mode.
- 3) Automatic switching between RX and TX mode.
- 4) Automatic work waiting timer supervision for use with T=0 protocol.
- 5) Enhanced speed SIM's as specified in GSM 11.11 - Phase 2+ using the SIM Baud Rate Factor settings.
- 6) Both SW and HW controlled T=0 protocol.
- 7) Work Waiting for T=0 mode.
- 8) 1 MHz SIM clock in low power mode.
- 9) GSM Phase 2 clock stop modes.
- 10) Automatic power down for immediate SIM deactivation.

The SIM card interface block (UART) can operate in two modes, named character mode and T=0 instruction mode respectively.

4.1.1 SIM Character Mode

In character mode the UART generates a character interrupt at the reception or transmission of every single character. In this configuration the UART is only capable of either sending or receiving one character at a time before SW intervention is needed, therefore the protocol needed to establish the communication with the SIM Card must be handled solely by software: this also involves procedure bytes handling during the instruction execution. The only logic functionality provided by the UART is the ability to retransmit characters with parity errors and to signal parity error detected in reception. The behavior of the character mode is illustrated in Figure 4-2.


Figure 4-2

Author	Enrico Bandera	Department:	S2	Page:	4/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

At the mandatory transmission speed (F=372 and D=1) this will cause a worst-case interrupt density of one interrupt per 1.37ms during instruction execution. Having to support high speed SIM transmission (F=512 and D=8) the interrupt load would increase by almost a factor of 7. At this higher speed the interrupt density would become one interrupt per 236µs in character mode, which would be a drastic increase of the SIM originating interrupt load.

4.1.2 SIM T=0 Protocol Mode

As a result of the increased interrupt load at high transmission speed in character mode, the T=0 instruction mode has been implemented as a HW block in the EGoldRadio SIM module. Such HW T=0 controller is capable of handling the T=0 protocol with a minimum SW intervention. Besides providing retransmissions and parity error signaling it also handles all procedure bytes. Moreover it can be easily combined with a PEC channel for transferring data to and from the SIM Card transparently to the MCU, leaving just a very few interrupts to be served by the SW. The HW controlled T=0 instruction execution is illustrated in Figure 4-3.

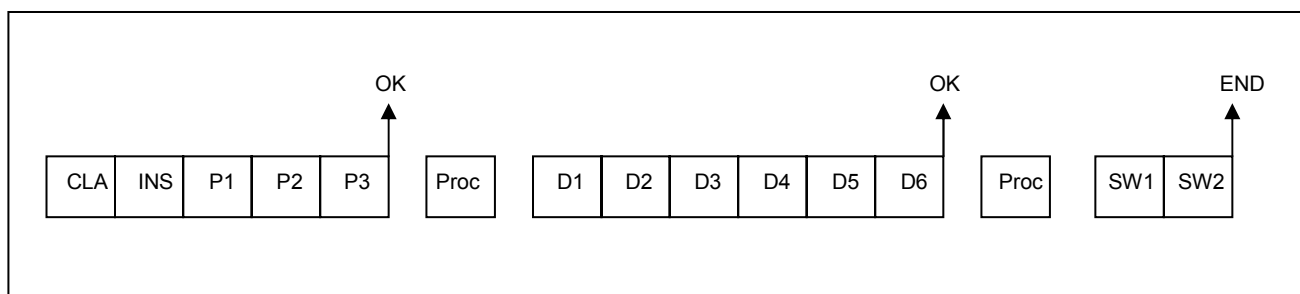


Figure 4-3

When a command is about to be executed the SW must set up the PEC to transfer the five command header bytes from the command store to SIMTX output register, then the T=0 controller is enabled and performs the data transfer without SW involvement. When then last of the five command header bytes has been transferred the PEC generates the first interrupt (UARTOK). The software must now re-program the PEC to either transfer data to SIMTX (Tx-command) or to receive data from SIMRX (Rx-command). Either way the HW T=0 controller will control the PEC again to perform the programmed data transfer and when the last character has been transferred by the PEC the second software interrupt (UARTOK) will be generated. In order to complete the instruction the software just needs to await the T0END interrupt, which is fired after reception of the status words, and the result can be read from registers SIMSW1 and SIMSW2.

4.2 Card sessions phases

The Chip Card Interface driver is capable of performing the activation and deactivation of a SIM Card in accordance with ISO/IEC 7816-3 and of executing APDU commands towards the SIM Card according to the T=0 transmission protocol. The different parts of the CI functionality are mutually exclusive during a card session, which is therefore divided into three phases, as illustrated in Figure 4-4.

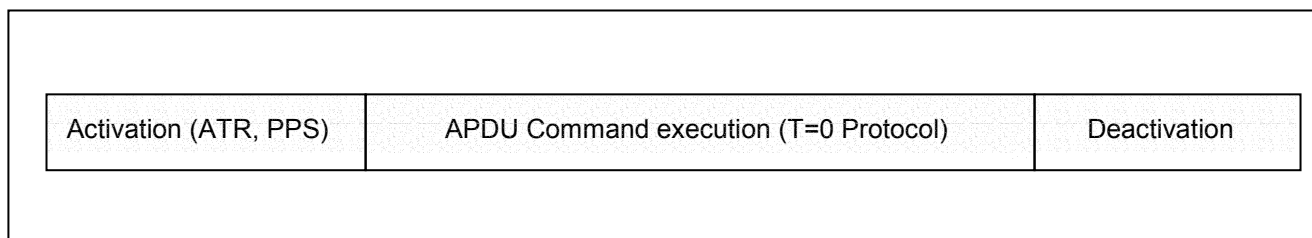


Figure 4-4

In the Activation phase the card session is started, which means that the physical signals towards the SIM are activated in the order specified by SO-IEC 7816-3. The Answer To Reset (ATR) characters is received and interpreted and if needed the Protocol and Parameter Selection (PPS) procedure is carried out. CI initiates the

Activation phase at power up. If no card is present or if the card cannot be activated, SI is informed and the card session ends. If on the other hand a SIM card is present and capable of being activated, SI is informed and the APDU command execution phase is entered. CI could also initiate the activation phase when a SIM is inserted after power up, but this option is disabled as the related PIN for SIM presence detection is not available in BP2 platforms.

In the APDU command execution phase CI is serving as a slave to SI. On request from SI, CI executes APDU instructions - one by one - towards the SIM over the T=0 transmission protocol. The APDU command execution phase is only reached when a SIM has been successfully activated. Since SIM cards can operate at a various different voltage levels and since some SIM cards only can indicate the supported voltage level after ATR and possible PPS (from the GSM directory parameters), there is a mechanism enabling SI to restart the SIM card at a different voltage. This basically means that SI can order CI to re-enter the Activation phase again if needed.

In the Deactivation phase the physical signals towards the SIM are deactivated in the specified order ending the card session. The Deactivation phase is entered when the SIM card is removed or when it is not possible to operate the SIM. The Deactivation phase can also be reached on the request of SI.

4.3 Software structure

The Chip Card Interface driver (CI) is divided into three abstraction levels of functionality as illustrated in Figure 4-5.

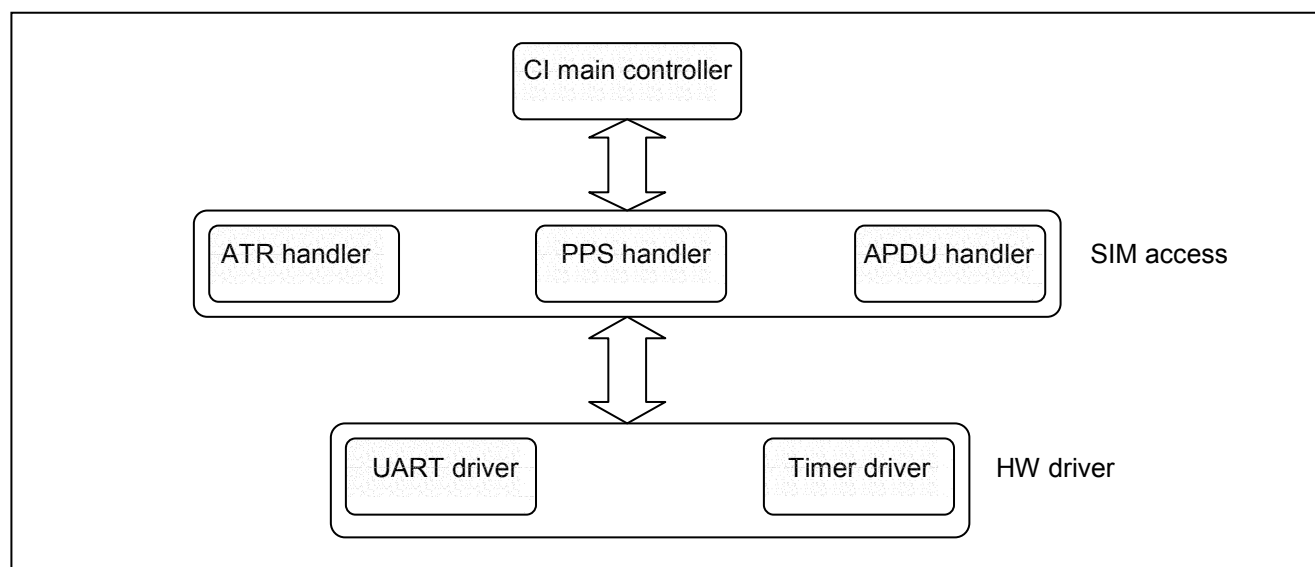


Figure 4-5

The main controller, which is responsible for the overall behaviour of CI, drives the three main engines: ATR handler, PPS handler and APDU handler to carry out the requested tasks towards the SIM. In turn these engines make use of the UART and TIMER drivers to control the interface related to their specific task.

The three main engines can only be operative one at a time. Therefore interrupts generated by the HW drivers (UART and TIMER interrupts) will be routed to the particular engine, which is active at the time the interrupt fires.

As usual for most of the drivers the finite state machines run in the context of a LISR: on the other side SDL Signals are sent to the MMI by a HISR, triggered by a LISR via a dedicated fast semaphore. The purpose of this architecture is to avoid overloading an interrupt service routine with time-consuming tasks such as allocating memory and sending signals.

4.4 Main controller

The main control state machine is implemented in CI_main_ctrl and is responsible for controlling the following tasks:

- 1) Activate the SIM contacts at power up or at card insertion and hereafter receive and interpret the Answer To Reset (ATR) characters.
- 2) Perform the Protocol and Parameter Selection (PPS) negotiation procedure - if needed.
- 3) Execute APDU commands towards the SIM – on request from SI.
- 4) Deactivate the SIM contacts at power down or at SIM removal or on request from SI.

4.4.1 Activation

Once CI has been started the main state machine tries to activate the SIM – through the ATR engine - at the lowest supported ME voltage level (which is 1.8V on the BP2 platform). The state machine is however able to switch to 3 Volts whenever needed.

Some SIM Cards are capable of indicating the supported voltages level within the ATR characters. If that is the case the main state machine will take this indication into account while activating the SIM, simply by using the indication to switch voltage level if possible. The activation sequence with respect to the voltage handling from the ATR characters is illustrated in Figure 4-6.

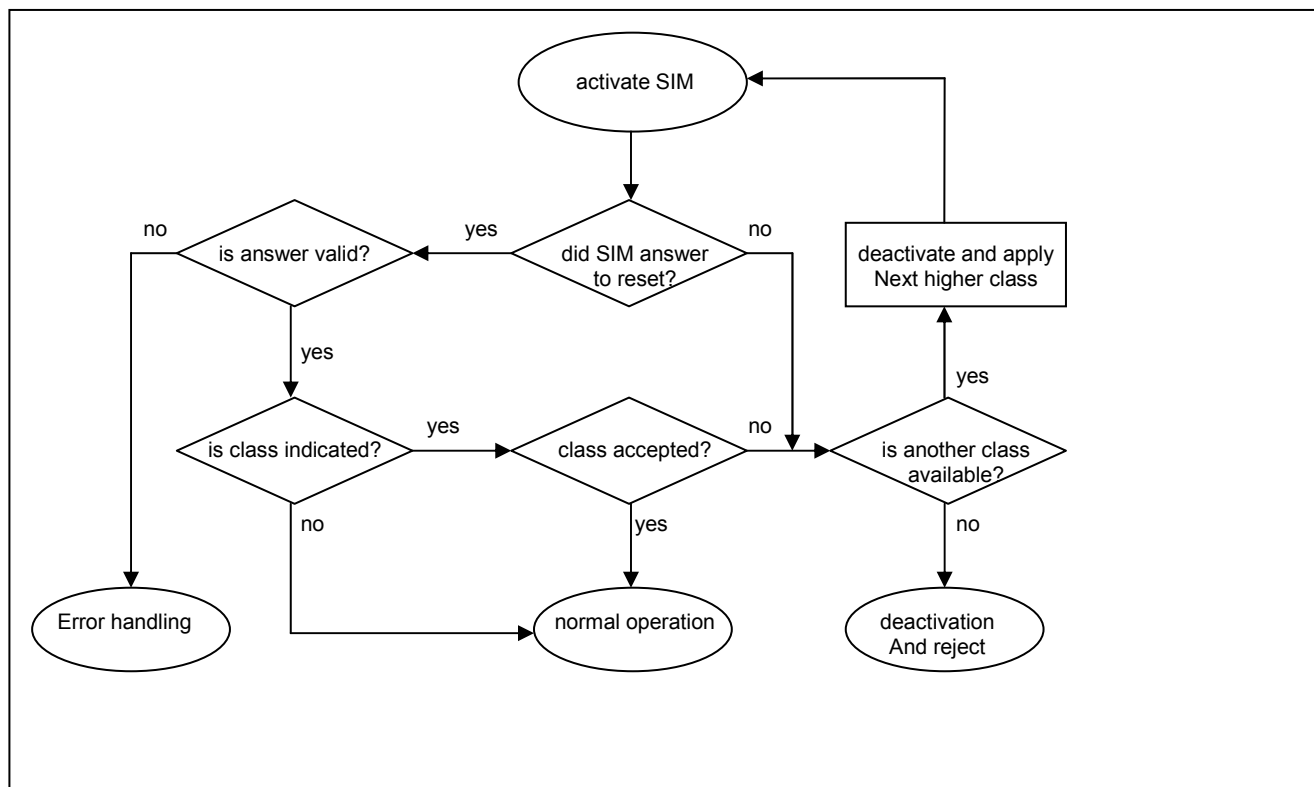


Figure 4-6

If the supported voltage level is not indicated in the ATR characters, it can be obtained from the GSM directory characteristic information in the “Execute APDU commands” phase. Since data of the APDU commands are completely transparent to CI, the supported voltage level will be given to CI by SI via the function SIM_SetCardPara. The return parameter of this function will inform SI whether or not the supported voltage is the same as the current used voltage. If there is discrepancy between the two values SI will initiate a reset of CI

Author	Enrico Bandera	Department:	S2	Page:	7/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential	

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

using the function SIM_ResetCard resulting in resetting the SIM. In other words the voltage switching procedure is indicated by CI but controlled (initiated) by SI.

Once the ATR engine has carried out the ATR sequence, the main state machine is informed of the activation result (succeeded or failed). If the activation was successfully carried out, the main state machine either carries out a PPS procedure as indicated by the ATR engine, otherwise CI will be ready to serve T=0 APDU command requests from SI.

According to GSM 11.11 the activation of the SIM Card will be repeated at least 3 times in case of failing or missing ATR from the SIM. Should all three ATR attempts fail if the ME is capable of operating at different voltage level(s) then it will retry activation at the next higher voltage level, otherwise the SIM Card will be rejected.

No matter the outcome of the ATR and possible PPS sequence, CI will send a status signal SICI_CCI_STATUS to SI indicating the current status of the SIM.

4.4.2 PPS negotiation

If the ATR state machine indicates that a PPS procedure is required, the main state machine will order the PPS engine to start the PPS procedure and will be informed when it has been executed and which is the result. If the PPS procedure was successfully executed, the values negotiated by the PPS engine will be used by the main state machine to program the UART to use the negotiated operating speed towards the SIM. CI is now ready to serve T=0 APDU command requests from SI. If the PPS procedure failed CI will repeat the activation procedure twice (3 times in all) as specified in GSM 11.11. The two first PPS requests will contain the desired values of F and D whereas the third PPS request will contain the normal speed values of F and D (GSM 11.11 specified feature). In order to make sure that the ME can handle a SIM Card where the PPS procedure has been wrongly implemented, the following safety mechanism has been implemented: if the PPS procedure fails after three PPS negotiation attempts, the SIM will be activated a 4th time without the PPS procedure. This will of course mean that enhanced transmission speed is not used (the ME will merely work as a phase 1 ME) but this backdoor will however allow the ME to operate the SIM after all.

4.4.3 APDU command execution

At this stage the SIM has been activated and the transmission speed has possibly been changed as a result of a PPS negotiation procedure. SI can now request CI to execute an APDU command using the function SIM_WriteCmd, which in turn starts the APDU engine by providing both the formatted command and the direction (either RX or TX). While the APDU engine is carrying out an instruction, no other command request will be accepted hence only one instruction can be executed towards the SIM at the same time.

Once the command has been successfully executed the main state machine is informed and the command result will be sent to SI in the signal SICI_SIM_RSPWRITE. If an error occurred then SI will receive a status signal SICI_CCI_STATUS instead of the command response signal.

4.4.4 Deactivation

The main state machine controls the deactivation of the SIM contacts. This is done at powerdown, removal of the SIM (not supported in BP2 platform) or simply on request from SI (SIM_DeactivateCard) and it basically means that the contacts towards the SIM are deactivated in the following order, according to ISO-IEC 7816-3: RST, CLK, I/O and finally Vcc.

4.4.5 Main control State machine

Figure 4-7 represents a simplified state diagram of CI main control state machine, including the main states and the most frequent events.

Author	Enrico Bandera	Department:	S2	Page: 8/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -		Confidential	

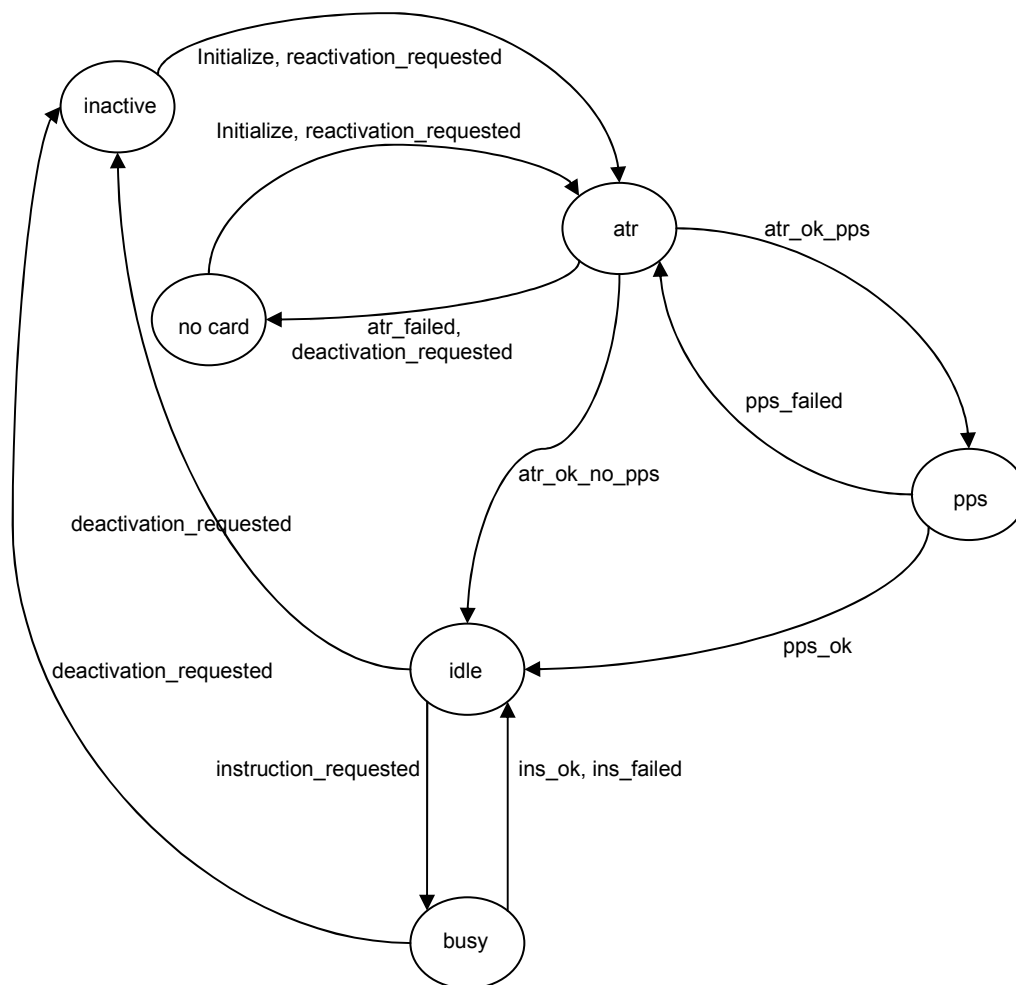


Figure 4-7

4.5 SIM access

The SIM access layer consists of three independent state machines (engines), which perform different tasks according to the current operational phase. These state machines are described in the following subchapters.

4.5.1 ATR handler

The ATR sequence is started at the command of the main control state machine. The first action carried out by the ATR state machine is the electrical SIM activation in the order Vcc, I/O, CLK and RST as described in ISO/IEC 7816-3. There are two ways to invoke a SIM card:

1. Internal Reset (IR) card, which means that the ATR is triggered by activating the SIM clock before RST is activated and therefore without reference to the RST signal. The new version of ISO/IEC 7816-3 (release 1997) does no longer cater for the IR activation form but it is nevertheless supposed to be backwards compatible with older SIM cards.
2. Active Low (AL) Reset card, which means that the ATR transmission by the card is triggered by the low to high transition of RST. AL is by far the most commonly used activation form in live SIM cards.

Author	Enrico Bandera	Department:	S2	Page:	9/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

	<h1>Technical Specification</h1>	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	----------------------------------	---

The ATR state machine CI_ctrl_atr is operating the SIM card interface in character mode, which means that the state machine receives an interrupt every time a character is received by the UART. Therefore the states of the ATR state machine more or less reflect the characters valid for an ATR sequence.

In order to make the ATR state machine operate in accordance with the transmission convention supported by the SIM-card, the parity error signalling must initially be disabled on the UART. As soon as the TS character is received the parity signalling is enabled (and remains enabled for the rest of the card session) and the content of the TS character instructs the further transmission convention (whether direct or inverse convention is supported).

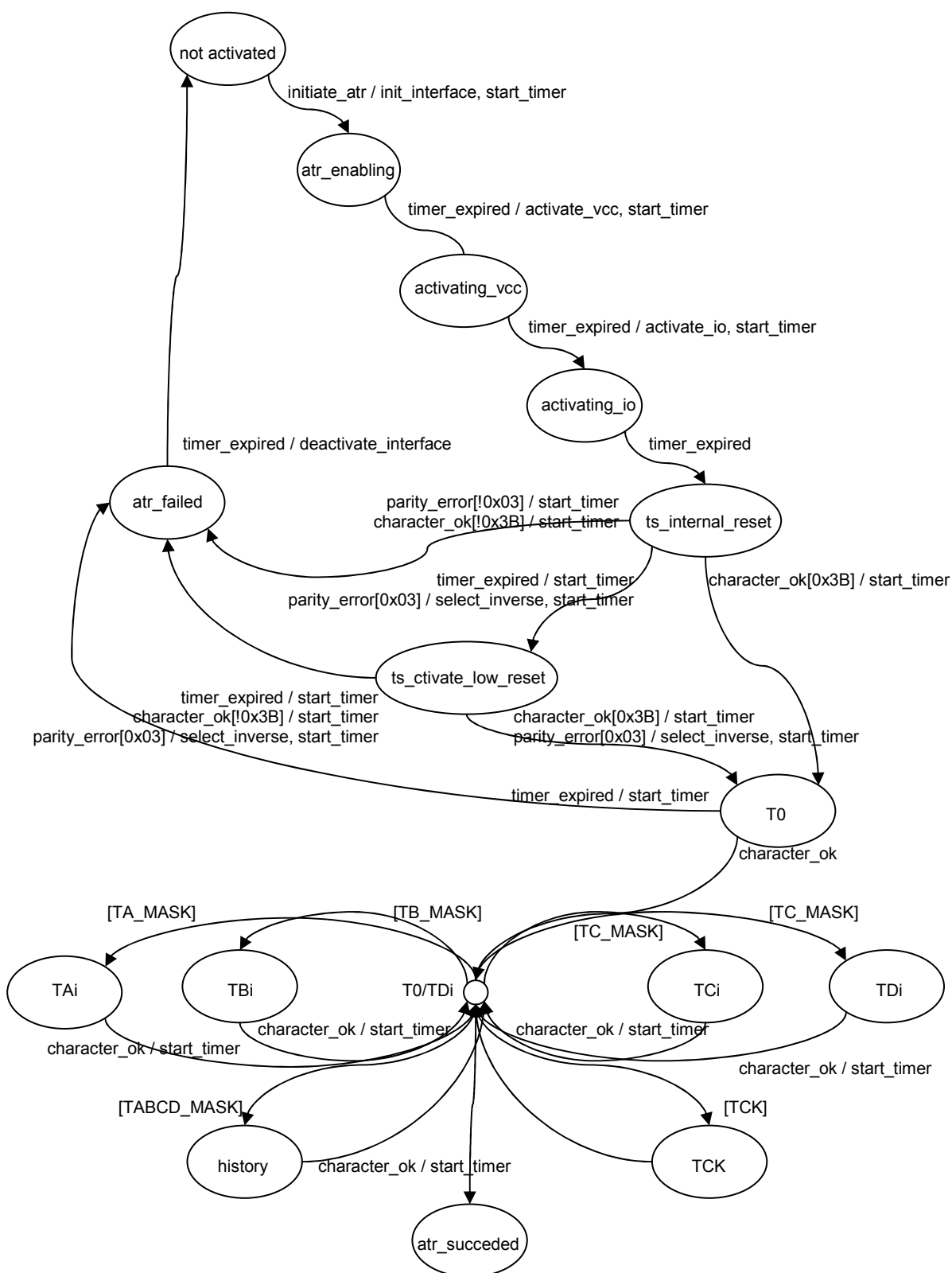
Once the correct transmission convention has been established, the data received from the ATR characters (if included in the ATR sequence) is processed as follows:

1. Extract the clock rate convention factor (FI) and the baud rate adjustment factor (DI) from ATR character TA(1). If the transmission factors are different from the default values (0x11 and 0x01), the main state machine shall be informed that a PPS procedure is required after the ATR sequence.
2. Check that TB(1) is 0. Otherwise reject the SIM.
3. Calculate the guard time from the extra guard time factor N given in TC(1).
4. Check if TD(i-1) for i>2 indicates T=15 protocol. If that is the case and if TA(i) following is present, extract the clock stop indication (XI) and the voltage class indication (UI) from TA(i). This information is used by the main control state machine to see if a restart at another voltage level is required.
5. Extract the parameter WI from TC(2) to calculate the required minimum Work Waiting Time.
6. Handle the unused interface characters.
7. Handle the historical characters.
8. Calculate, receive and compare – if required – the checksum character to detect whether an error had occurred or not. If the received checksum is invalid, the ATR sequence will fail.

Once the ATR sequence has ended (successfully or not), the main state machine is informed of the activation outcome. It is then the task of the main controller to act accordingly (either initiating a new activation sequence or a PPS procedure or simply being ready to execute APDU commands).

A simplified diagram of the finite state machine is represented in Figure 4-8: note that a timer expired in any of the states Tai, TBi, TCi, TDi, history and TCK causes a transistion to atr_failed, not depicted for lack of room.

Author	Enrico Bandera	Department:	S2	Page: 10/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential


Figure 4-8

Author	Enrico Bandera	Department:	S2	Page:	11/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

4.5.2 PPS handler

The PPS procedure is initiated by the main control state machine if the ATR state machine has notified the need for PPS. The content of what is sent to the SIM in the PPS request depends of what is suggested by the SIM in ATR characters. The PPS procedure is handled by the function `CI_ctrl_pps`.

When the UART is programmed to operate at a given transmission speed, this is done through the Baud Rate Factor register `SIMBRF`. The BRF can be computed from the formula:

$$BRF = \frac{F}{4 \times D}$$

In Table 4-1 all UART compliant combinations (integer values) of F and D are listed. The value written to `SIMBRF` register must be an integer therefore not all combinations of F and D are acceptable. If the SIM is suggesting one of the acceptable F and D pair then a PPS request is issued using these values. Should the SIM on the other hand suggest an unsupported combination of F and D, the ME can either suggest another pair in the range $F_d < F < F_i$ and $D_d < D < D_i$ or simply stick to the default values $F = F_d$ and $D = D_d$.

Howether since real SIM cards today only supports the default values and the mandatory high speed values of F and D, the ME will for the time being only support these two combinations, which are shaded in Table 4-1. The reason for this choice is that the EGoldRadio interface has not been thoroughly tested against other additional enhanced transmissions speeds.

Once the PPS request has been sent to the SIM, the SIM shall return a PPS response. The result of the PPS procedure is indicated to the main controlling state machine that will act accordingly (use default values, use negotiated values or reactivate the SIM).

Author	Enrico Bandera	Department:	S2	Page: 12/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential

f[MHz]	F	D	BRF	Bitrate [bit/s]	T _{WETU} [μs]
3.25	372	1	93	7,737	114.46
	512	1	128	6,348	157.53
		2	64	12,695	78.77
		4	32	25,391	39.38
		8	16	50,781	19.69
		16	8	101,563	9.85
		32	4	203,125	4.92
	744	1	186	4,368	228.94
		2	93	8,737	114.46
	768	1	192	4,232	236.29
		2	96	8,464	118.15
		4	48	16,927	59.08
		8	24	33,854	29.54
		12	16	50,781	19.69
		16	12	67,708	14.77
		32	6	135,417	7.38
	1024	1	256	3,174	315.06
		2	128	6,348	157.53
		4	64	12,695	78.77
		8	32	25,391	39.38
		16	16	50,781	19.69
		32	8	101,561	9.85
	1116	1	279	2,912	343.41
	1488	1	372	2,184	457.88
		2	186	4,368	228.94
		4	93	8,737	114.46
		12	31	26,210	38.15
	1536	1	384	2,116	472.59
		2	192	4,232	236.29
		4	96	8,464	118.15
		8	48	16,927	59.08
		12	32	25,391	39.38
		16	24	33,854	29.54
		32	12	67,708	14.77
	1860	1	465	1,747	572.41
	2048	1	512	1,587	630.12
		2	256	3,174	315.06
		4	128	6,348	157.53
		8	64	12,695	78.77
		16	32	25,391	39.38
		32	16	50,781	19.69

Table 4-1

A simplified diagram of the finite state machine is represented in Figure 4-9.

Author	Enrico Bandera	Department:	S2	Page:	13/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential	

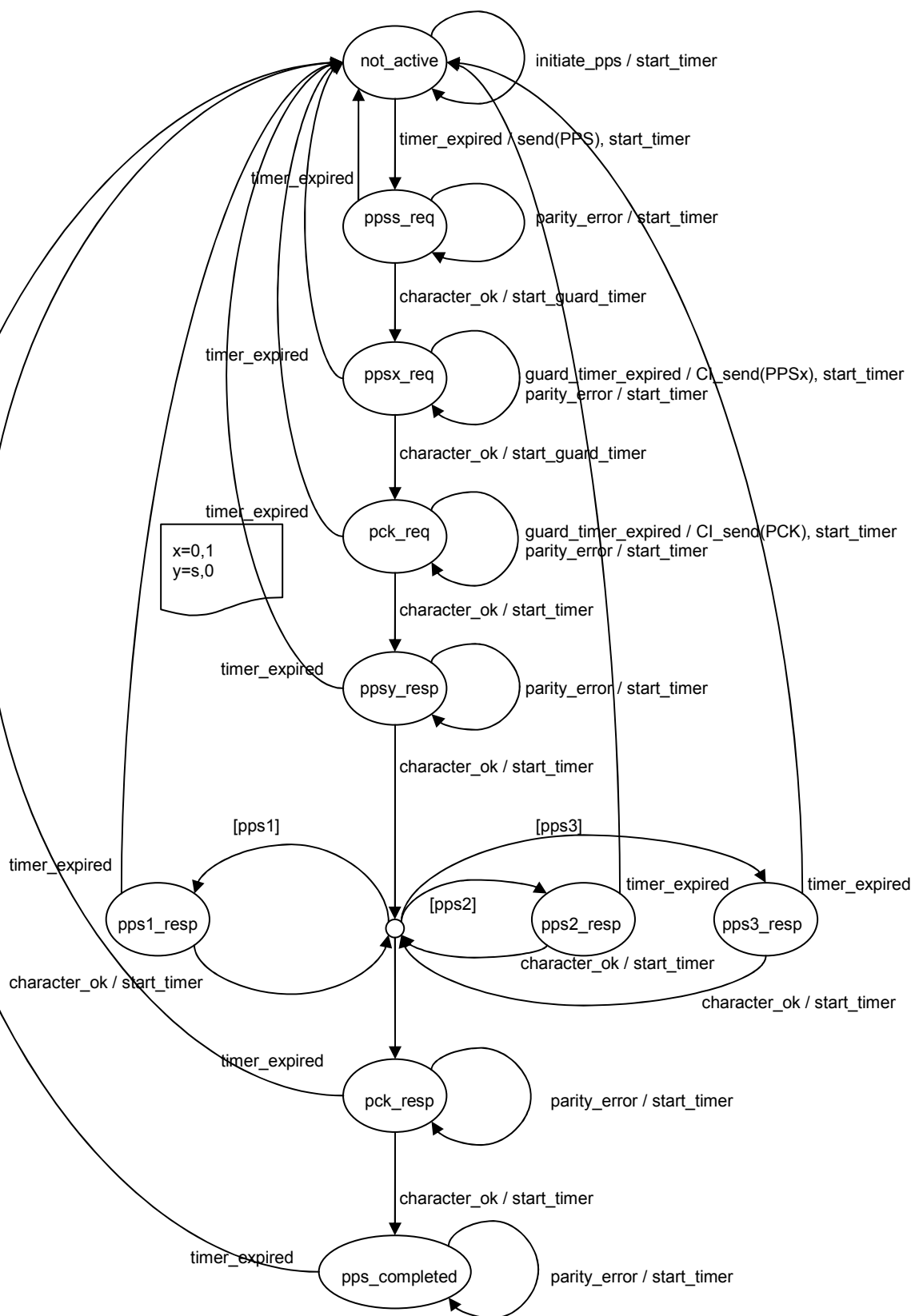


Figure 4-9

Author	Enrico Bandera	Department:	S2	Page:	14/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

4.5.3 T=0 instruction handler

The T=0 instruction handler (also named APDU handler) CI_ctrl_ins is responsible for executing APDU commands - one by one - towards the SIM. This state machine makes use of the hardware implemented T=0 controller, which handles character retransmission, parity error signalling and procedure byte management. Combining the HW T=0 controller and the PEC, the interrupt load is narrowed down to 3 SW interrupts (or 4 interrupts for data size larger than 254 bytes) for each instruction (see Figure 4-3).

The T=0 instruction handler will initiate the command execution by setting up the PEC to transfer 5 bytes (the command header bytes) before enabling the HW T=0 controller. When the 5 header bytes have been transferred by the PEC (to SIMTX), the PEC generates the first interrupt (UARTOK) to the T=0 instruction handler. The T=0 instruction handler must now re-program the PEC to transfer the command data, from the command store to SIMTX for transmission and from SIMRX to the command response store for reception. Once such transfer is complete the PEC generates a second interrupt (UARTOK) to the T=0 instruction handler. If the data size is larger than 254 bytes the PEC must once again be reprogrammed to handle the last one or two data bytes, then T=0 instruction handler will receive another UARTOK interrupt. The HW T=0 controller finally handles the reception of the status words SW1 and SW2, after which it generates the last interrupt (T0END) indicating to the T=0 instruction handler that the command has been successfully executed (at least from the T=0 protocol point of view).

The execution result is then indicated to the main controlling state machine that is responsible for sending a response to SI.

Before executing any instruction if the SIM Clock is stopped then it must be re-started. According to GSM 11.11 the ME should wait at least 744 clock cycles before initiating the command after having activated the clock. The instruction state machine ensures this delay and the clock is then afterwards active throughout the entire command execution.

When the instruction has been executed, GSM 11.11 prescribes that the ME shall wait at least 1860 clock cycles before switching off the clock. Now, commands are very likely to be issued in bursts involving several commands and it is therefore a waste of time to switch off the clock after every issued command. For this reason the state machine simply waits 1-2ms after executing a command. As a result the clock will be active if a new command request should be received within this interval. The clock stop level (high or low) is indicated from SI using the function SIM_SetCardPara.

Older SIM cards does not support that the clock is stopped during a card session. Instead of stopping the clock, the lower clock of 1.08MHz will be used instead. Anyway 1.8 and 3V SIM cards must support clock stop.

A simplified diagram of the finite state machine is represented in Figure 4-10.

Author	Enrico Bandera	Department:	S2	Page: 15/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential

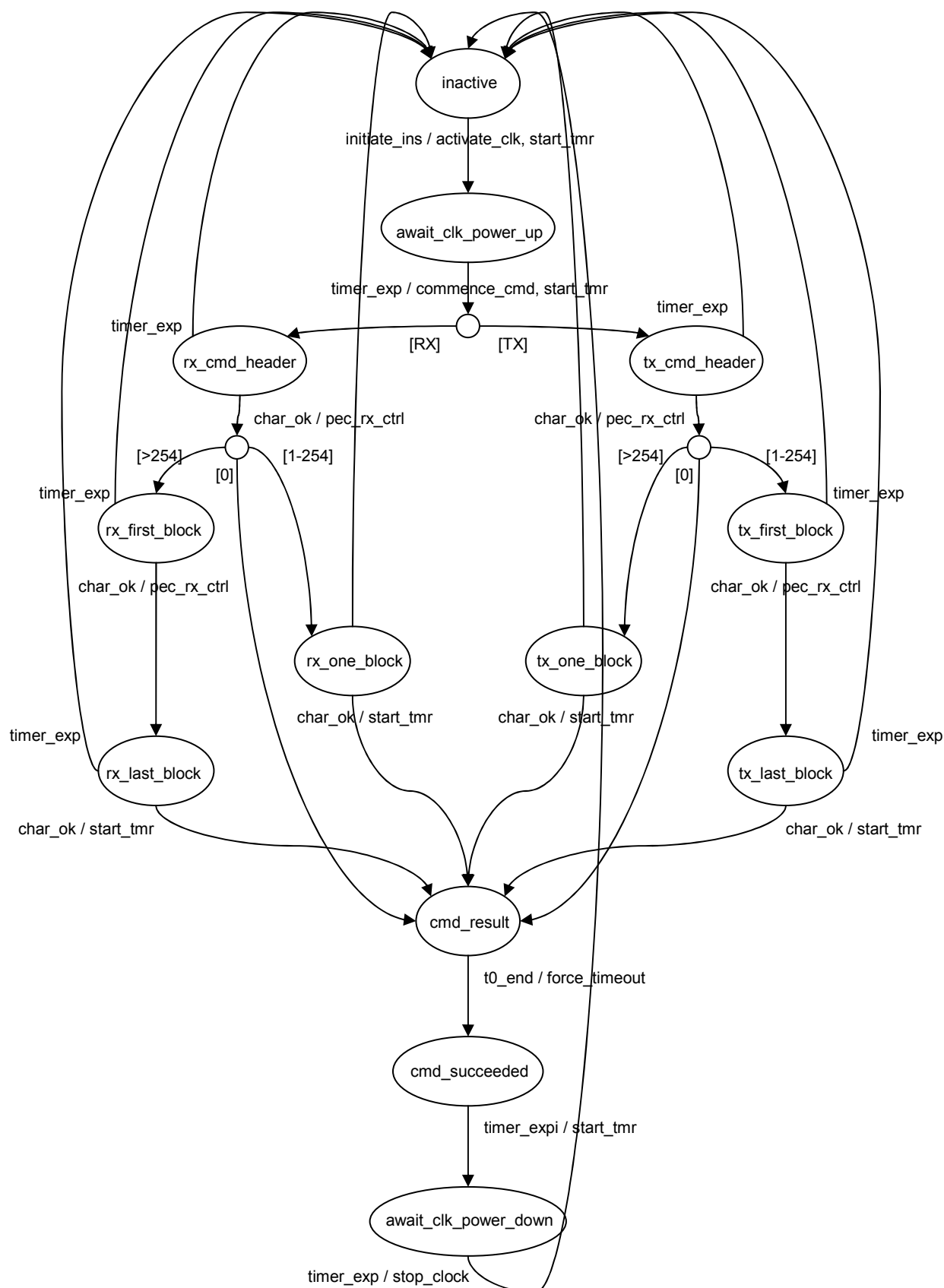


Figure 4-9

Author	Enrico Bandera	Department:	S2	Page:	16/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

4.6 HW drivers

The SIM access layer functions are gaining access to the EGoldRadio HW blocks using two drivers, described in the following subchapter: the former contains functions for controlling SIM Card interface device (the UART) and the latter for managing the low level timer.

4.6.1 UART driver

The half-duplex UART driver provides the following functionalities by means of a synchronous interface based on function calls:

1. Initialise the UART and related interrupts.
2. Transmit a single character on the serial I/O line to the SIM.
3. Receive a single character on the serial line from the SIM.
4. Enable and disable automatic parity error indication to the SIM.
5. Select transmission convention (direct or inverse).
6. Select operating voltage (either 1.8V or 3V)
7. Switch supply and programming voltage (Vcc and Vpp) to the SIM on and off.
8. Set the reset signal to the SIM to either high or low level.
9. Switch the SIM clock signal either on or off.
10. Enable and disable the I/O line.
11. Control the clock stop mode and level (stop at either high or low level or use 1.08 MHz clock).
12. Control of HW implemented T=0 protocol.
13. Enabling/disabling of the HW T=0 state machine.
14. Read out the status words SW1 and SW2 at the end of the T=0 instruction.
15. Control the bit rate used in the communication with the SIM.

The following registers are used:

SIMID	SIM Identification Register
SIMCTRL	SIM Control Register
SIMBRF	SIM Baud Rate Factor Register
SIMSTATUS	SIM Status Register
SIMIRQEN	SIM Interrupt Enable Register
SIMTX	SIM Transmit Register
SIMRX	SIM Receive Register
SIMINS	SIM Instruction Class Register
SIMP3	SIM Parameter 3 Register
SIMSW1	SIM Status Word 1
SIMSW2	SIM Status Word 2
SIMRXSPC	SIM RX command SPace
SIMTXSPC	SIM TX character SPace

For the meaning and usage of the SIM interface registers please refer to EGoldRadio documentation.

4.6.2 Timer driver

The timer driver offers the features listed below:

1. Initialise the timer and related interrupt.
2. Start a timer of variable length meeting the timer requirements of ISO/IEC 7816-3 for signal spacing.
3. Starting a guard timer with fixed length (PPS request character spacing).
4. Stop a previously started timer.
5. Handle the expiration of a previously started timer.

The timer handling is achieved by using a dedicated HW timer in the EGoldRadio. CI has been assigned to use the General Purpose Timer T2. The following registers are necessary to implement the timers:

Author	Enrico Bandera	Department:	S2	Page:	17/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

T2CON	Timer 2 Control Register
T2	Timer 2 Register
T2IC	Timer 2 Interrupt Control Register

The T2 timer duration is calculated from the following formula:

$$f_{T0} = \frac{f_{HW}}{BPS1 * 2^{T2I}}$$

The resolution of the timer in order to fulfil the requirements of ISO/IEC 7816-3 with respect to high-speed data transfer rate must be 19,7µs. The T2CON prescale parameter T2I should thereby be 101b such that the T2 prescaler is 256.

The timer is controlled by a 16 bit register and the maximum timer duration is therefore found in the following formula:

$$Duration = Resolution * FFFFh = 19,7\mu \times FFFFh = 1.29s$$

Now since the maximum required time duration to control the SIM is larger than the maximum time duration supported by T2, it is necessary to implement a SW-controlled prescaler.

5 Interface

This chapter contains the interface description of the SIM driver, with particular attention to the operative interface towards the Subscriber Identity Module Application (SI) but including also the production test interface towards ATC and the power down interface towards the idle task. As usual the interface from a client to the driver is synchronous (via function calls) while the interface from the driver to the client (SI in particular) is asynchronous (based on SDL Signals).

5.1 SI interface

The primary user of CI is the Subscriber Identity Module Application part (SI), which is using CI to execute commands towards the SIM Card. CI provides a number of services to SI, which allows SI to control the behavior of CI.

The communication from SI to CI is based on a pure function call interface. Whenever CI needs to inform SI about the occurrence of a given event (presence/removal of SIM or APDU command execution response), this is done by use of SDL signals. This means that the interface from CI to SI is entirely based on signals.

5.1.1 SI Synchronous interface

Prototype

SDL_Void SIM_WriteCmd (sim_cmdDir_type sim_cmdDir, SDL_Integer sim_lgth, T_BYTE_PTR sim_cmdPtr)

Functional description

This function is used by SI to initiate an APDU command towards the SIM card. If SI has already initiated an APDU command and this command has not yet been completed then SI is not allowed to initiate another command using this function. It is the responsibility of SI to fill in unused or RFU command data bytes (e.g. CHV1 verification command data with a PIN equal to "1234" must be: 0x31 0x32 0x33 0x34 0xFF 0xFF 0xFF 0xFF where the last four 0xFF padding bytes are generated by SI).

Parameters

- **sim_cmdDirection:** holds the APDU command transmission direction (either transmission e.g. UPDATE BINARY or reception e.g. GET RESPONSE).
- **sim_cmdLgth:** holds the APDU command length (the number of bytes). The command has a minimum mandatory length of 5 bytes (the command header) and an additional optional length of up to 255 bytes (command data).

Author	Enrico Bandera	Department:	S2	Page:	18/22
Filename	SIM_Driver_Specification.doc				
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG -			Confidential	

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

- **sim_cmdPtr:** holds a pointer to the first byte of the APDU command store. The command store will be available during the entire command execution and must include both the 5 command header bytes and the optional command data bytes. Note that P3=0 in Tx instruction mode introduces no command data to be transferred whereas P3=0 in Rx instruction mode introduces 256 bytes response data to be transferred. For instructions carrying neither Rx nor Tx data (e.g. INVALIDATE), the requested instruction must be set to Tx (because P3=0).

Return value

None

Prototype

SDL_Boolean SIM_SetCardPara (sim_voltage_type sim_voltage, sim_clkStop_mode_type sim_clkStop_mode)

Functional description

This function is used by SI to transfer a part of the GSM/DSC1800 directory characteristics to the SIM driver, which uses the information to determine if a voltage switch is needed and to control the SIM clock during the card session.

Parameters

- **sim_voltage:** The SIM supported voltage characteristics (5v/3v/1.8v/) as indicated in the GSM/DCS1800 directory information.
- **sim_clkStopMode:** The SIM supported clockstop characteristics (no stop/no preferred level/low level/high level) as indicated in the GSM/ DCS1800 directory information.

Return value

TRUE if the current operating voltage level is different from the one indicated in the GSM/DSC1800 directory information, and a re-activation at the appropriate voltage level is therefore needed (SI must initiate a re-activation of the SIM using the function SIMReset); FALSE if the current operating voltage is in line with the value e indicated in the GSM/DSC1800 directory information, so that the card session can continue.

Prototype

SDL_Void SIM_ResetCard(SDL_Void x)

Functional description

This function is used by SI to force a re-activation of the SIM (e.g. if operation is needed at another voltage level or in case of a proactive SAT Refresh). The term re-activation refers to a deactivation of the SIM followed by a new activation of the SIM and again followed by ATR and possible PPS handling.

Parameters

None

Return value

None

Prototype

SDL_Void SIM_DeactivateCard (SDL_Void x)

Functional description

This function is used by SI to force a deactivation of the SIM, removing all signals according to ISO-IEC 7816-3.

Parameters

None

Return value

None

5.1.2 SI Asynchronous interface

Name

SICI_CCI_STATUS

Description

This signal is used by CI to signal a state change, for example due to an unsuccessful instruction execution, to invalid data received from the Card or to an instruction request while another one is already in progress.

Parameters

- **Param1:** carries the new SIM status: inserted, removed, error, busy.

Name

Author	Enrico Bandera	Department:	S2	Page: 19/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

SICI_SIM_RSPWRITE

Description

This signal is sent by CI when an APDU command has been successfully executed and carries the command result.

Parameters

- Param1: Holds the APDU response length (the number of bytes). The response has a minimum mandatory length of 2 bytes (status words SW1 and SW2) and an additional optional length of up to 256 bytes. The maximum APDU response length is therefore 258 bytes.
- Param2: Holds a pointer to the first byte of the APDU response buffer. Note that SW1 and SW2 are located as the two last bytes of the APDU response store (SW2 as the last). If the APDU command has no response data, the buffer only contains SW1 and SW2.

5.2 ATC interface

The interface towards ATC is used in the production line to test if the hardware related SIM interface has been properly assembled. The interface between ATC and CI is prely synchronous and basically consists of one function returning the present status of the CI – and thereby the status of the SIM-card.

Prototype

ci_sim_status_type CI_GetSIMStatus(void)

Functional description

This function is used by ATC during production test to detect whether the SIM-card interface is operational or not.

Parameters

None

Return value

- not_ready: this value is returned if the status request is received while activating the SIM (before it is possible to determine if there is a card or not).
- no_sim_present: this value is returned when CI has detected that there is no connection towards an inserted SIM-card.
- sim_present: this value is returned when the inserted SIM-card can be activated by CI – meaning that the hardware connections seem to be working.

Prototype

SDL_Boolean CI_EnterTestMode(void)

Functional description

This function is used by ATC during production test to set CI in a special test mode where CI - upon normal activation - will inform SI that no SIM is present (also if a SIM-card actually is inserted). Furthermore CI wil let ATC use the function CI_GetSIMStatus in order to check if the SIM connection is operational.

Parameters

None

Return value

TRUE whether the SIM driver successfully entered Test Mode, FALSE otherwise.

5.3 Power down interface

The interface towards the Idle Task is used to identify when CI is ready to accept that the EGoldRadio is powered down (that is the 13MHz is stopped). CI will not allow power down while performing commands towards the SIM or during ATR and PPS procedure.

Prototype

SDL_Boolean CI_PowerDownAllowed(void)

Functional description

This function is used by the idle task to determine whether CI can accept power down of the 13 MHz clock or not.

Parameters

None


Author	Enrico Bandera	Department:	S2	Page: 20/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

Return value

TRUE if CI can accept power down, FALSE otherwise.

Author	Enrico Bandera	Department:	S2	Page: 21/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential

	Technical Specification	Doc. ID: AH01.SW.TS.000012 Rev.:2.0 Date:05/01/2006
---	--------------------------------	---

6 References

6.1 External

GSM 11.10: "Digital cellular telecommunications system (Phase 2); Mobile Station (MS) conformance specification"

GSM 11.11: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment (SIM – ME) interface"

ISO/IEC 7816-3: "Identification cards – Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols"

GSM 11.12: "Specification of the 3 Volt Subscriber Identity Module - Mobile Equipment (SIM - ME) interface"

GSM 11.18: "Specification of the 1.8 Volt Subscriber Identity Module - Mobile Equipment (SIM - ME) interface"

E-GOLDradio PMB 7870 GSM/GPRS Single Chip Solution - Design Specification, Rev 1.05 2005-08-02 - Section 10.8 (SIM interface)

6.2 Internal

None.

7 Document change report

	Change Reference		Record of changes made to previous released version	
Rev	Date	CR	Section	Comment
1.0	01/06/2004	N.A		Document created
2.0	05/01/2006	N.A.		Updated to BP30 platform with E-GoldRadio chip

8 Approval

Revision	Approver(s)	Date	Source/signature
1.0	Stefano Godeas	01/062004	Document stored on server
2.0	Valter Jelcic	05/01/2006	Document stored on server

Author	Enrico Bandera	Department:	S2	Page: 22/22
Filename	SIM_Driver_Specification.doc			
M06-N7 Rev. 2	Copyright (C) 2006NeonSeven S.R.L. All rights reserved - Exclusive property of Infineon Technologies AG –			Confidential