IFI CAN_FD IP (Flexible Data-Rate)

User Guide

see also IFI_CANFD_regs.pdf for register description

Core Version:	0x09115B15
Document Version:	2.1
Document Date:	12. 2017

Important Notes

- *In April 2012 BOSCH released the revision 1.0 of the CAN FD standard*
  - *for flexible Data Rate*
- *In 2014 this standard was converted to ISO 11898-1*
- *For improving the residual error-rate there are a few changes included into the ISO standard, which make both versions incompatible to one another*
  The older standard is called non-ISO, the newer ISO
- *To give our customers the possibility to switch between both standards we organized the registers in a way that*
  - *customers can use the old register map for the (old) BOSCH standard called „non-ISO“ and*
  - *the new register map for the new „ISO“ standard (signed with § in this document)*
- *To switch between both standards customers have to set or clear the ena_ISO bit in the CMD register*
  - *! before ! writing all the control words*
- *In February 2015 we added a new Configuration Bit Bit 26, so customers can switch the Bittiming independend from the ena_ISO bit*
- *ALTERA® is now part of Intel®, so references to Altera in this document should be read as references to Intel*
- *The IFI CAN-FD IP-core revision 0x09115B15*
  passed the ISO CAN conformance tests on December 13th, 2017
  According to "ISO 16845-1:2016 Road vehicles - Controller area network (CAN) - Conformance test plan" and C&S enhancement/corrections according to "CAN CONFORMANCE TESTING Test Specification C&S Version r3d08"

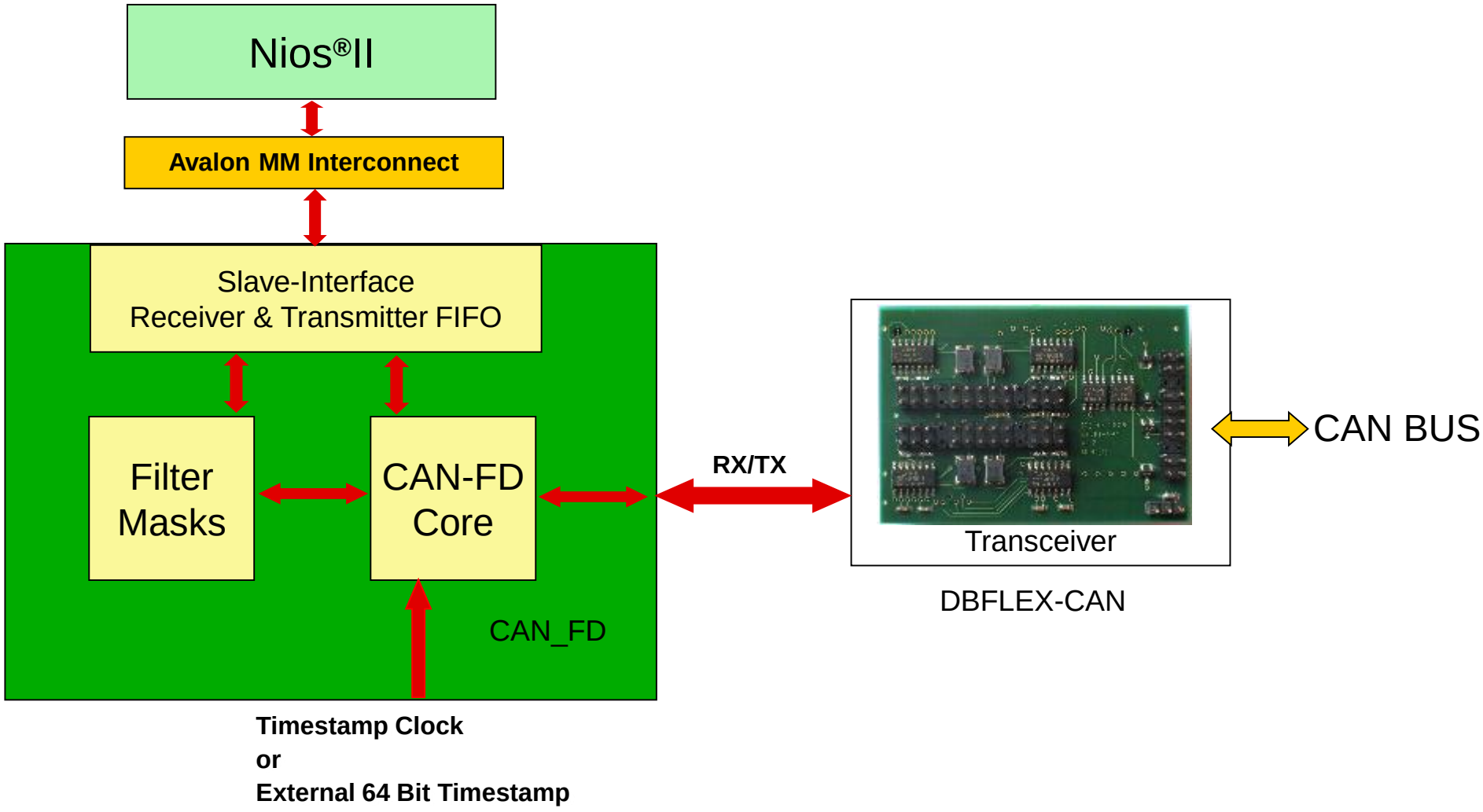
I/F/I F. Sprenger

Contents

Overview

- *Block Diagram*
- *Feature List*
- *Altera Implementation*
- *OpenCore Plus Feature*
- *Reference Design*
- *Using SignalTap II*
- *Pricing*
- *References*
- *CAN Background*
- *Contacting Technical Support*

Block Diagram



IFI CAN_FD Feature List

- CAN 2.0B
  - Standard or Extended Identifier
  - Remote Frames
  - Error-Handling
- CAN FD ISO and non-ISO
  - up to 64 Byte Data
  - Flexible Data rate
- Separate Message FIFO for Receive / Transmit
  - each can be configured 1, 2, 4, 8, 16, 32 or 64 kBytes
  - dynamic size for each message to save memory resources
- 256 Message Filters
  - Every Message Filter contains one MASK- and one Identifier Register
- NIOS Interface
  - Example software included
  - HAL drivers for NIOS II included
- Separate Clock for CAN and Avalon-IF possible
- One High Priority Message
- Bus-Statistic possible
- 32 Bit Timestamp (64 bit when Timestamp from external):
  - for received messages
  - possible for transmitted messages with frame number other than 0
- For external CPU support :
  - 8,16, 32 or 64 Bit Interface
- Reading of the compile time parameters possible

I/F/I F. Sprenger

IFI Feature Details

- Timestamp
  - “OFF” disabled Timestamp, saves about 100 LE
  - “ON_internal” Timestamp with 1 us resolution, TriggerPosition selectable SOF or ACK
  - “ON_External” External 64 bit Timestamp used
  - “ON_with_ExternalControl” for Clock and Reset for TimestampCounter, additional outputs for custom Timestamp

- Dual_Clock
  - “OFF” system-clock is used for CAN-timing, saves about 350 LE, CAN-timing may be limited because of the usually slower system-clock
  - “ON” separate Clock for system and CAN-timing, means with a faster CAN-Clock a higher Fast Baudrate can be achieved, optimal Clockrate for CAN-clock can be selected depending on Baudrate in external PLL

- Bus_statistic
  - “OFF” disabled Bus_statistic, saves about 300 LE
  - “ON” selectable Bus_statistic value in Total Frames or Bus_load in %

- Filter with 256 Mask+Identifier entries
  - “OFF” disabled filter, saves about 200 LE and 2 M9k
  - “ON” filter for ID, 11bit only ID, 29bit only ID, CAN-FD only ...

IFI Feature Details

- ID Format
  - “IFI_legacy”, identical to IFI CAN IP and IFI Advanced CAN IP
  - “CANalyzer”, identical to Vector Informatik CANalyzer
  - “ID_other”, 3rd ID format used in the CAN market
  -
- Additional Errorcounters for CAN_FD debug
  - count errors during arbitration phase if transmitter (slow Baudrate)
  - count errors during data phase if transmitter (fast Baudrate)
  - count errors during arbitration phase if receiver (slow Baudrate)
  - count errors during data phase if receiver (fast Baudrate)

IFI Additional Feature Details

- Programmable Transmission Suspend
  - 0 => no suspend, start next transmission as soon as valid (default)
  - <value> => suspend next transmission after a successful transmission or after abort when maximum repeat count is reached in <value> microseconds
  - global value or per individual message

- Programmable Transmission RepeatCount
  - 0 => endless repeat of transmitted message until successful transmission (default)
  - 1 => just one try => single shot mode, no repeat after error (used for MilCAN for example)
  - 2 => repeat ones, in case first try was not successful
  - <value> => make maximum <value< attempts to transmit message
  - global value or by individual message

- Bus_monitoring Mode
  - CAN_FD spec 3.3.1

- Restricted Operation Mode
  - CAN_FD spec 3.3.2

- Loopback Mode
  - internal or external

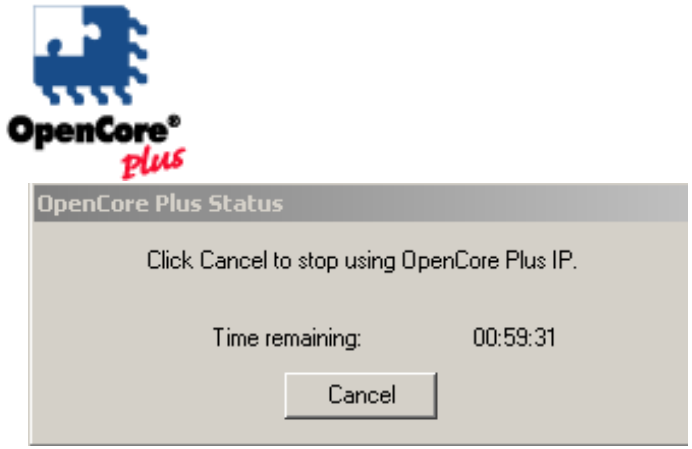
- Hard-Reset
  - CAN-core can be resetted hard

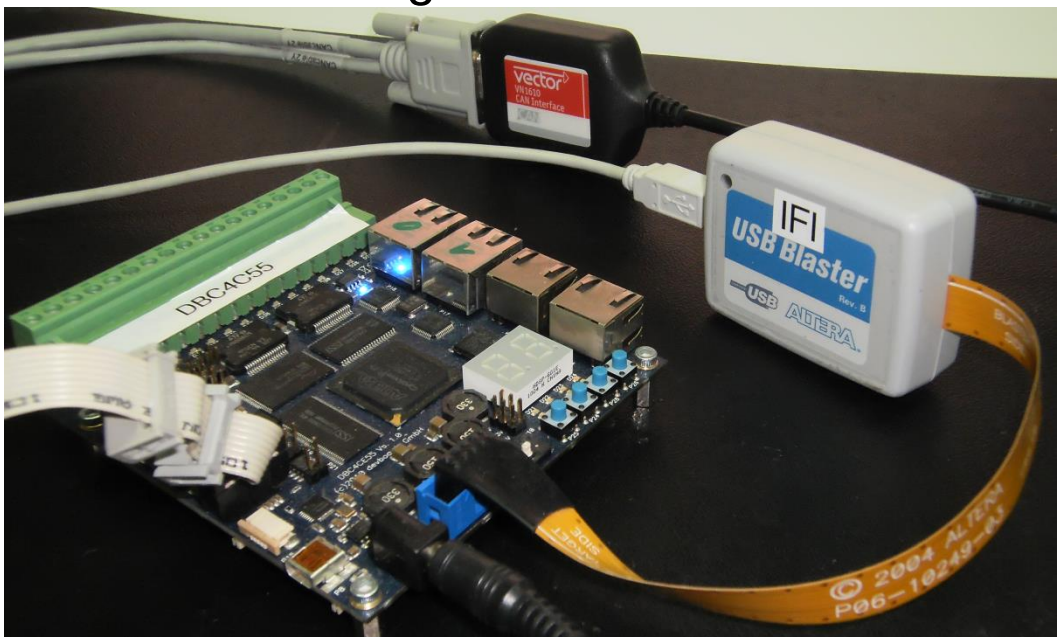
IFI CAN_FD Altera Implementation

- Design flows supported by
  - SOPC Builder
  - QSYS Interconnect
  - Platform Designer
- Device families targeted
  - CYCLONE I, II, III, IV, 10
  - MAX 10
  - CYCLONE V
  - STRATIX I,II,III,IV,V,10
  - ARRIA I,II,V,10
- Resources depend on the QuartusII version, compiler settings and options
- Device resource utilization
  - 4300 .. 5200 LEs CYCLONE I, II, III, IV
  - 4300 .. 5200 LEs MAX 10
  - 1900 .. 2550 ALM CYCLONE V
  - 2000 .. 2600 ALM STRATIX
  - 1900 .. 2500 ALM ARRIA
- Memory device resource M9k
  - minimum without filter 5 M9k
  - minimum with filter 7 M9k
  - maximum depend on the user setting Fifo size
- Memory device resource M20k
  - minimum without filter 3 M20k
  - minimum with filter 5 M20k
  - maximum depend on the user setting Fifo size

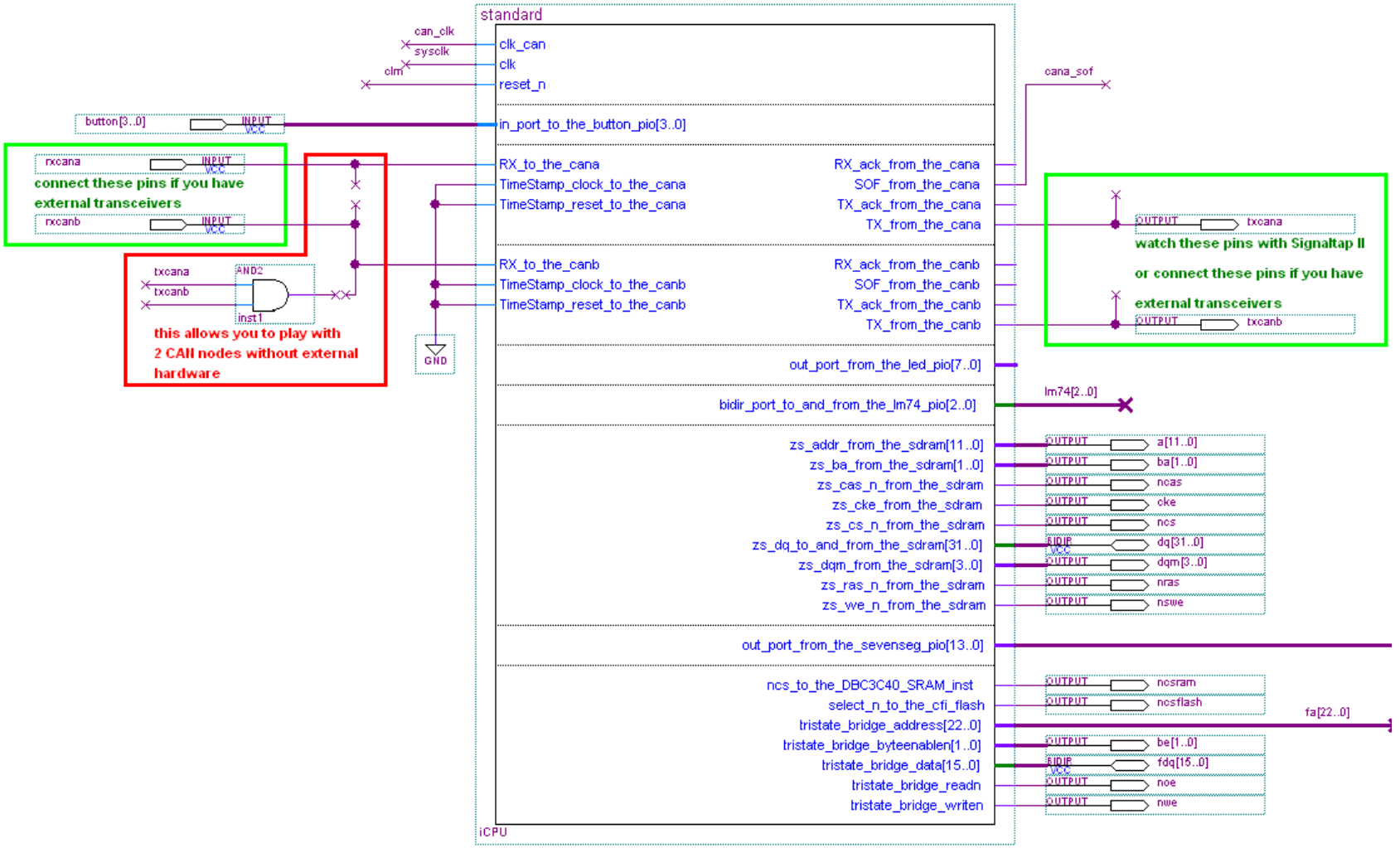
OpenCore Plus Feature

- Test the CAN_FD Module on your board or on development boards from ALTERA or devboards.de
  - There is no time limit with an established connection between the device and the Quartus programmer.
  - If you remove the connection the time remaining is ~ 1 hour.



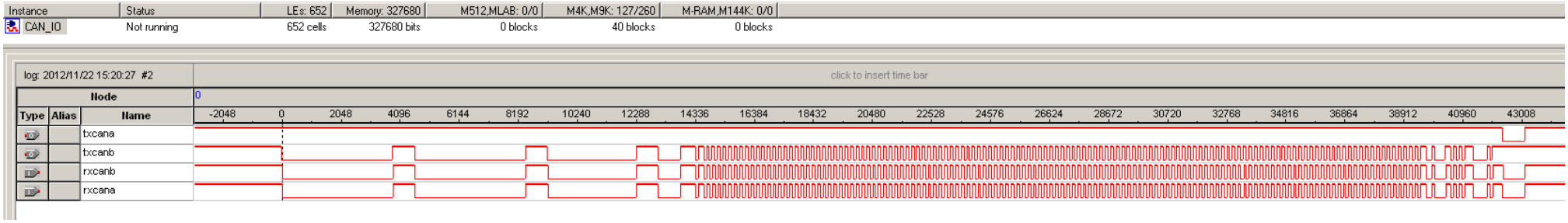


IFI CAN_FD Reference Design



- Test the IFI CAN_FD with / without external hardware

IFI CAN_FD Reference Design



- Use Signaltap II to watch the TX + RX pins

IFI CAN-FD Pricing (encrypted netlist)

- Node-Locked License: 9900.- €
  - 1 year maintenance included
  - T-Guard or NIC-ID Fixed Node
  - Maintenance: 990.- € / year
- Floating License: 12375.- €
  - 1 year maintenance included
  - Single or Multi Server
  - Maintenance: 1237.- € / year
- Licensing:
  - Unlimited (no timelimit) License, Multiproject,
  - Royalty Free with IFI
  - The CAN-NETWORK-PROTOCOL-License is not included
    - => Available by Bosch
    - search www for "Bosch_CAN_Protocol_License_Conditions"

IFI CAN-FD References

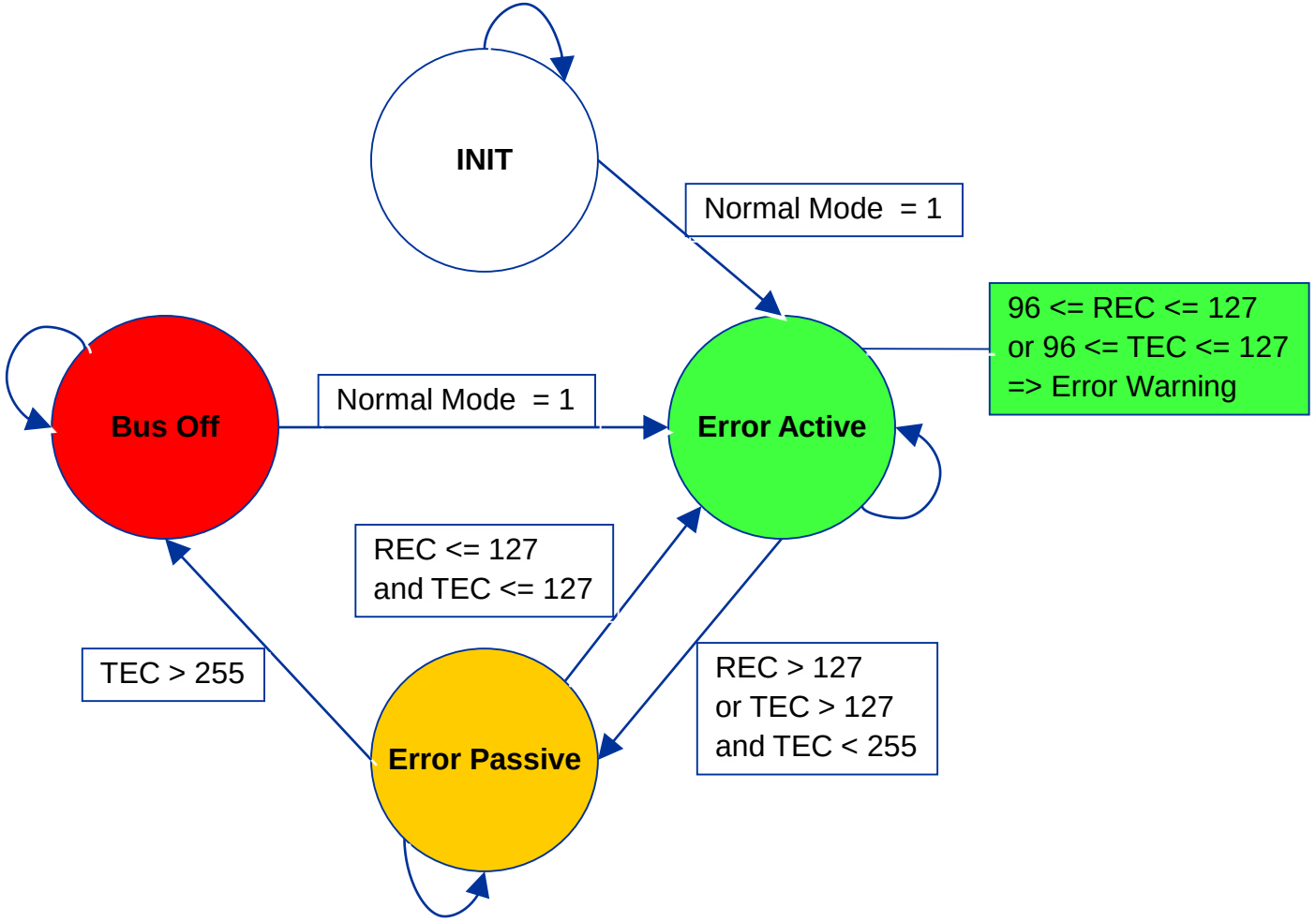
- Hardware tested on
  - CAN FD plug fest 2014 at CIA (CAN In Automation, Nuremberg) non-ISO
  - several plug fest in 2015 and 2016 ISO

- The IFI CAN-FD IP-core revision 0x09115B15
- passed the ISO CAN conformance tests on December 13th, 2017
- according to "ISO 16845-1:2016 Road vehicles - Controller area network (CAN) - Conformance test plan" and C&S enhancement/corrections according to "CAN CONFORMANCE TESTING Test Specification C&S Version r3d08"

CAN-Background

- **CAN Messages:**
  - Every CAN message consists of a certain number of bits that are divided into fields. There are fields such as the Arbitration Field, the Data Field, CRC, the End of Frame…
  - The Arbitration Field is different for CAN 2.0 A and CAN 2.0 B messages. It's a logical address with 11 bits for CAN 2.0 A and 29 bits for CAN 2.0 B. The lowest value is the highest priority = 0.
  - The Data Field contains the application data of the message with 0 to 64 bits (0 to 8 bytes) (0 to 64 bytes with CAN-FD).
  - With exception of the CRC delimiter, the ACK field and the EOF, the bits are stuffed. That means, 5 consecutive bits with identical value are followed by a complementary bit.
  - The Error Frame and the Overload Frame are of a fixed form and not coded with bit stuffing.
- **Error Detection:**
  - The error management unit is able to detect five different error types.
    - Bit Error
    - Bit Stuffing Error
    - CRC Error
    - Form Error
    - ACK Error
- **Error Handling:**
  - Error detected
  - Transmitting of error frame
  - Message will be discarded
  - Error counters are incremented
  - Transmission will be repeated
- **Error Limitation:**
  - To prevent a permanently disturbed bus each CAN controller has three error states.
    - Error Active
    - Error Passive
    - Bus Off

CAN-Error States



Contacting Technical Support

Although we have made every effort to ensure that this SOPC Builder Ready / QSYS OpenCore Package works correctly, there might be problems that we have not encountered. If you have a question or problem that is not answered by the information provided in this README file, please contact the IP Vendor or Intel (Altera is now part of Intel)

For questions about the core's features, functionality and parameter settings please contact:

IFI Ingenieurbüro Für Ic-Technologie
Franz Sprenger
Kleiner Weg 3
97877 Wertheim
Germany
Phone: (+49) 9342 96080
E-Mail: ifi@ifi-pld.de
http://www.ifi-pld.de

Installation

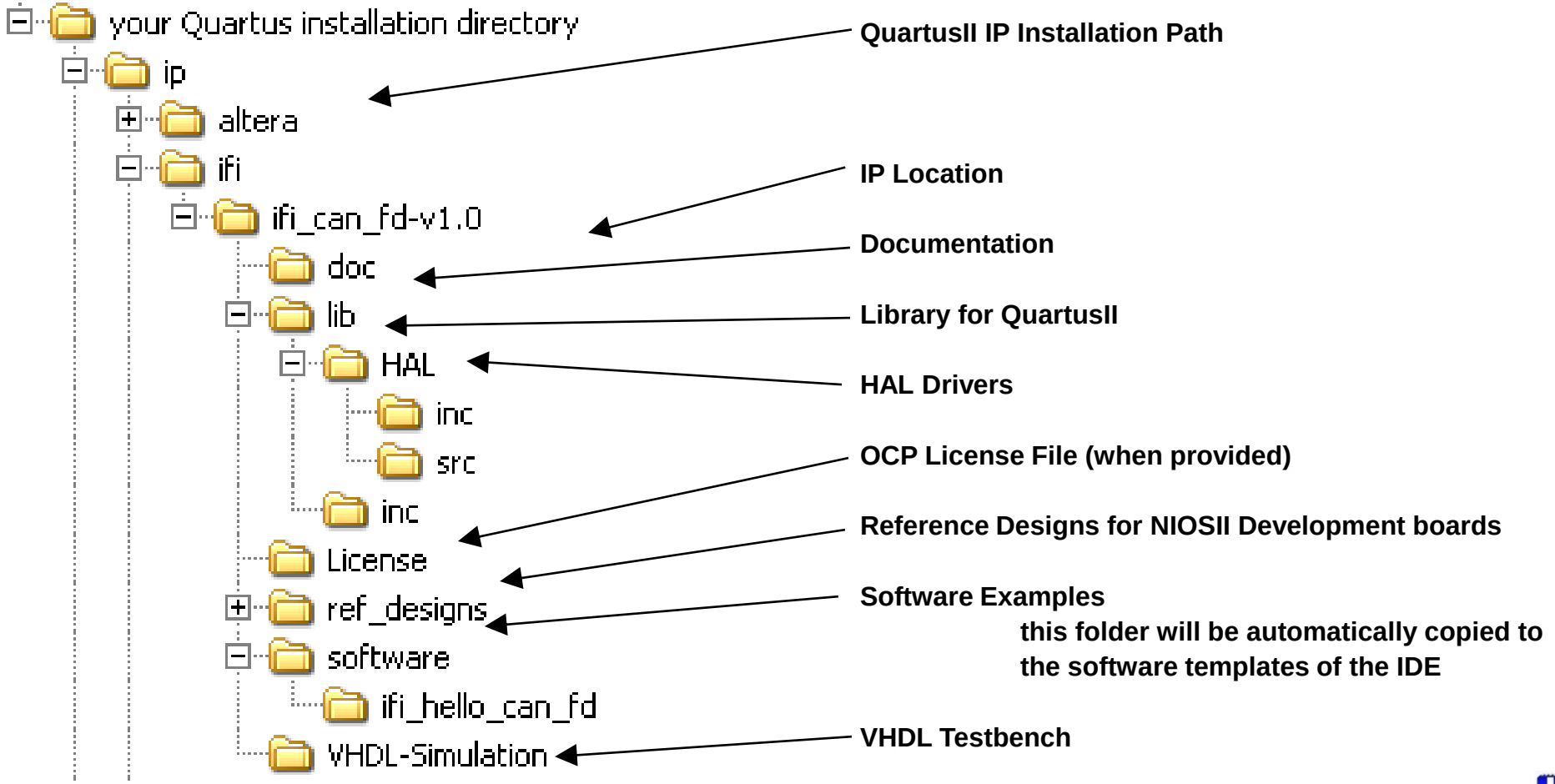
- *How to install*
- *SOPC Builder Ready OpenCore Package*
- *Licensing*
- *Set up Licensing*

Install the IFI CAN_FD

- Before you can start using IFI CAN_FD functions, you must install the IFI CAN_FD files on your computer. The following instructions describe this process.
- **Close QuartusII and IDE / SBT**
- **The installed Quartus II version has to be 9.1 or a newer version**
- **Install the IFI CAN_FD files**
  - The following instructions describe how you install IFI CAN_FD on computers running the Windows, Linux or Solaris operating systems.
  - If you don't change the installation path, the SOPC Builder and the Megawizard will find the IP automatically
  - Windows
    - Follow these steps to install the IFI CAN_FD on a PC running a supported version of the Windows operating system:
    - Choose Run (Windows Start menu).
    - Type <path name>\<filename>.exe, where <path name> is the location of the downloaded IFI CAN_FD function and <filename> is the filename of the IFI CAN_FD function.
    - Click OK. The IFI CAN_FD Installation dialog box appears. Follow the on-screen instructions to finish installation.
  - Solaris & Linux
    - Follow these steps to install the IFI CAN_FD on a computer running supported versions of the Solaris and Linux operating systems:
    - Decompress the package by typing the following command:
    - gzip -d<filename>.tar.gz
      - where <filename> is the filename of the IFI CAN_FD function.
    - Extract the package by typing the following command:
      - tar xvf <filename>.tar

SOPC Builder Ready OpenCore Package

The SOPC Builder Ready OpenCore Package contains all files required for plug-and-play integration of this core into Altera's SOPC Builder tool, allowing the user to easily evaluate the core within his Avalon-based system.



Licensing

OpenCorePlus License

The download of this IP-core is completed by an individual OpenCorePlus license named license_CAN_FD_ocp.dat or similar

When the FEATURE line from this license is appended to the user's Quartus II license file, the encrypted files can be read into Quartus II and place and route can be performed.

The license permits the generation of <revision_name>_time_limited.sof files.

The hardware evaluation feature will run during you have an established JTAG connection between your board and the QuartusII programmer or SignalTap. If you close the programmer it will stop working immediately. If you remove the connection it will stop working after 1 hour.

(Refer to the messages created by the programmer)

Full License

If you purchased a FULL LICENSE you receive an additional license file, license_???.dat.

Use this instead of the license_ocp.dat. When the FEATURE line from this license is appended to the user's Quartus II license file, the encrypted files can be read into Quartus II and place and route can be performed. The license permits the generation of <revision_name>.pof files and gate-level simulation netlists.

- One FEATURE line can span more than one line

Set Up Licensing

- To install your license, you can either append the license to your **license.dat** file or you can specify the IFI CAN_FD's **license_ocp.dat** file in the Quartus II software.
  - Before you set up licensing for the IFI CAN_FD, you must already have the Quartus II software installed on your computer with the licensing set-up.
- **Append the license to your license.dat file**
  - To append the license, follow these steps:
  - Open the IFI CAN_FD license file in a text editor.
  - Open your Quartus II **license.dat** file in a text editor.
  - Copy all lines from the license file and paste it into the Quartus II license file.
  - Do not delete any FEATURE lines from the Quartus II license file.
  - Save the Quartus II license file.
    - When using editors such as Microsoft Word or Notepad, ensure that the file does not have extra extensions appended to it after you save (e.g., **license.dat.txt** or **license.dat.doc**). Verify the filename in a DOS box or at a command prompt. Also, make sure that the file is saved in plain-text format without formatting characters.
- **Specify the license file in the Quartus II software**
  - To specify the IFI CAN_FD license file in Quartus II, follow these steps:
  - Altera recommends that you give the file a unique name, e.g., <*core name*>_**license.dat**.
  - Run the Quartus II software.
  - Choose **License Setup** (Tools menu). The **Options** dialog box opens to the **License Setup** page.
  - In the **License file** box add a semicolon to the end of the existing license path and filename.
  - Type the path and filename of the IFI CAN_FD function license file after the semicolon.
    - Do not include any spaces either around the semicolon or in the path/filename.
  - Click **OK** to save your changes.

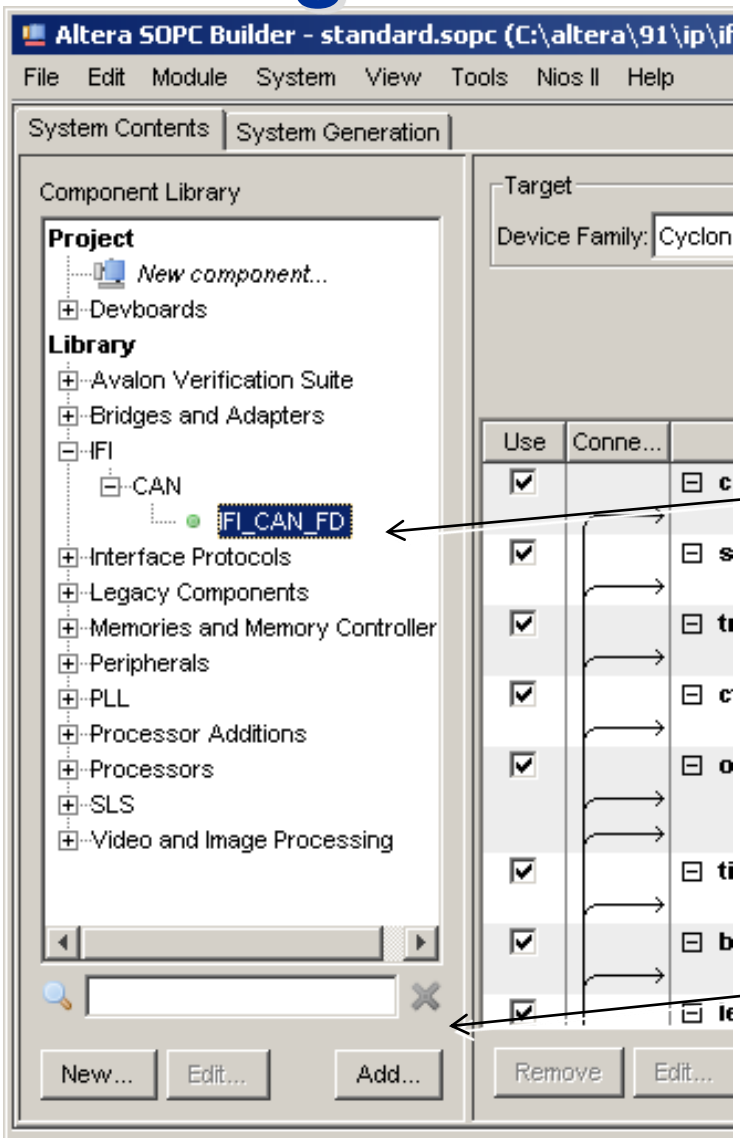
Integrating the Core using SOPC Builder

- *Adding the Core to your SOPC System*
- *About*
- *Documentation*
- *Parameterize*

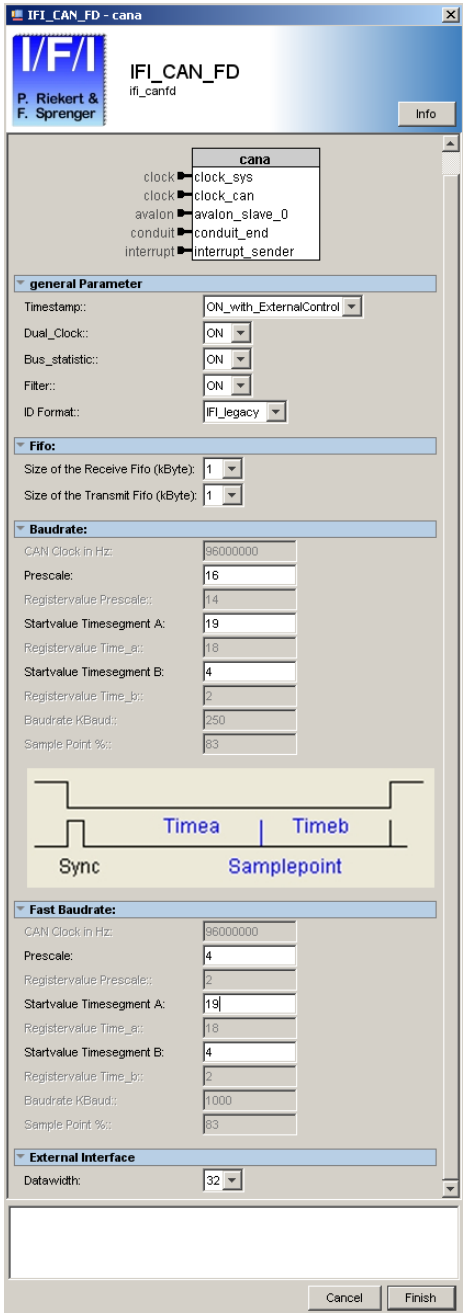
Integrating the Core into your System using SOPC Builder

- This section contains instructions on the following:
  - Adding the Core to your System
  - Running the Reference Design
- These instructions assume that the user is familiar with the
  - Altera OpenCore evaluation process,
  - Altera's Quartus II development software,
  - and the Altera SOPC Builder tool
- For more information on these prerequisites, please visit www.altera.com

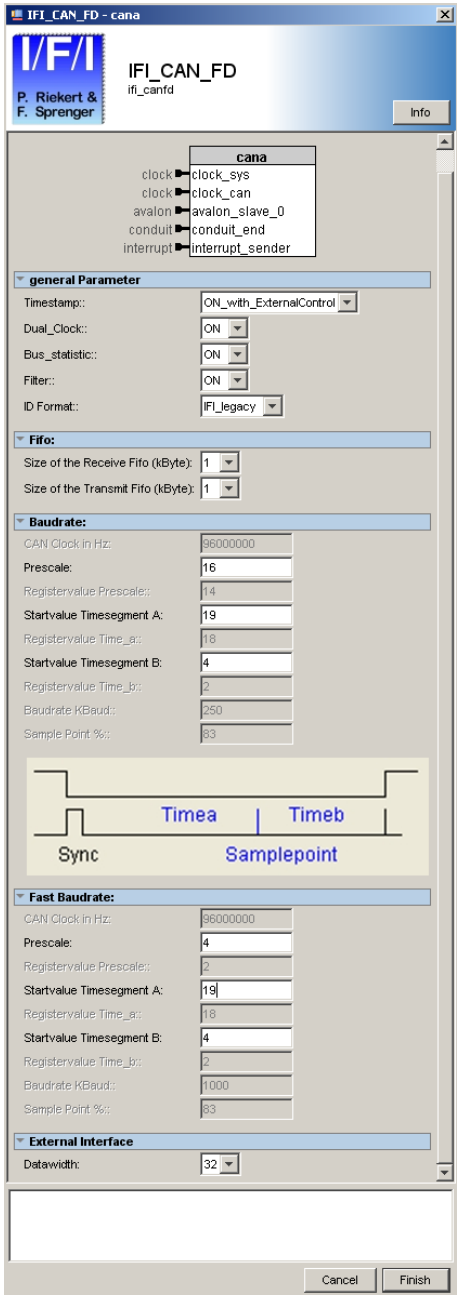
Adding the Core to your SOPC System

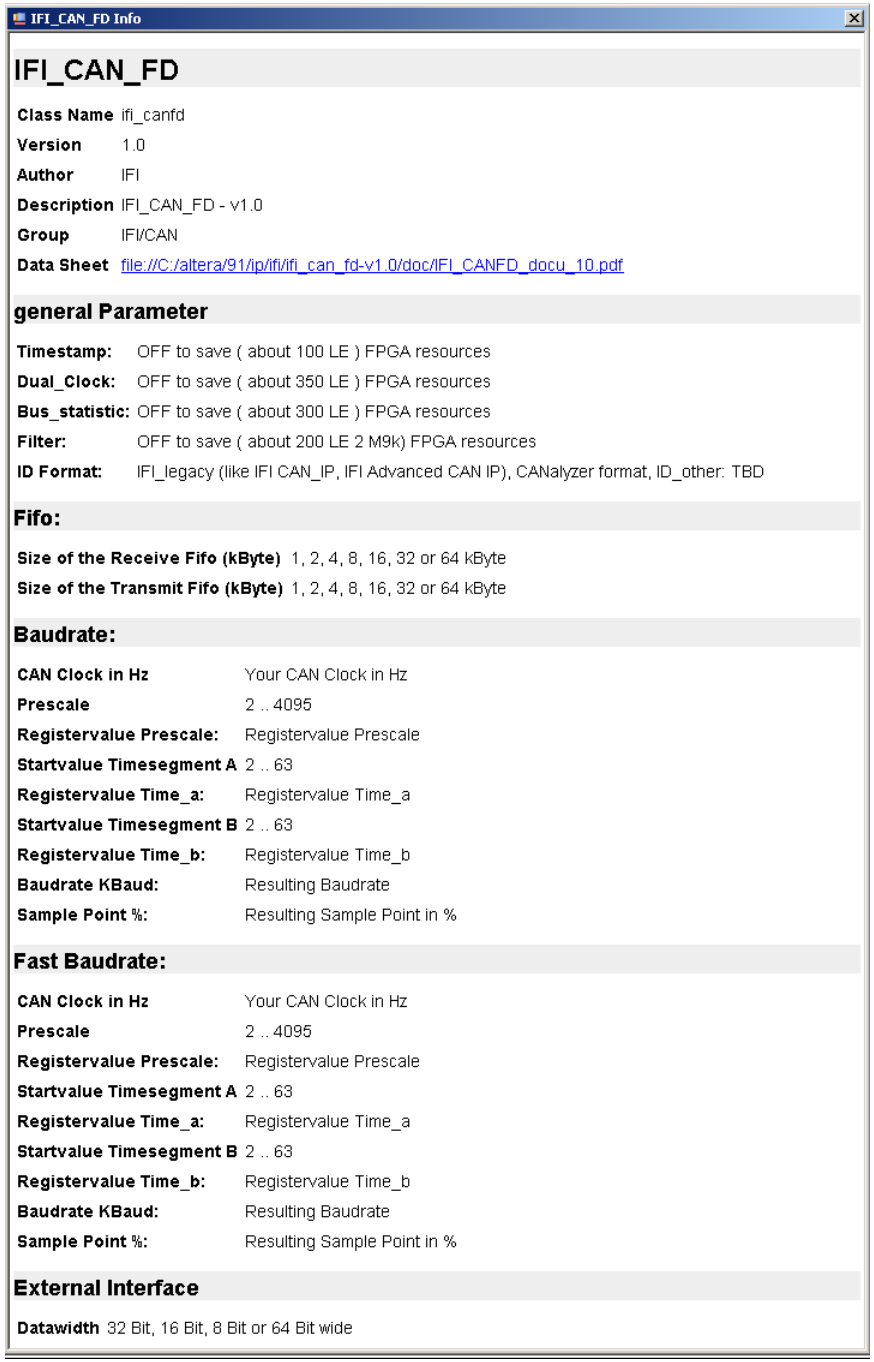


- Launch SOPC Builder from Quartus II (Tools menu).
- Select the core by clicking on the core name
- Click "Add" to add the core to your system.

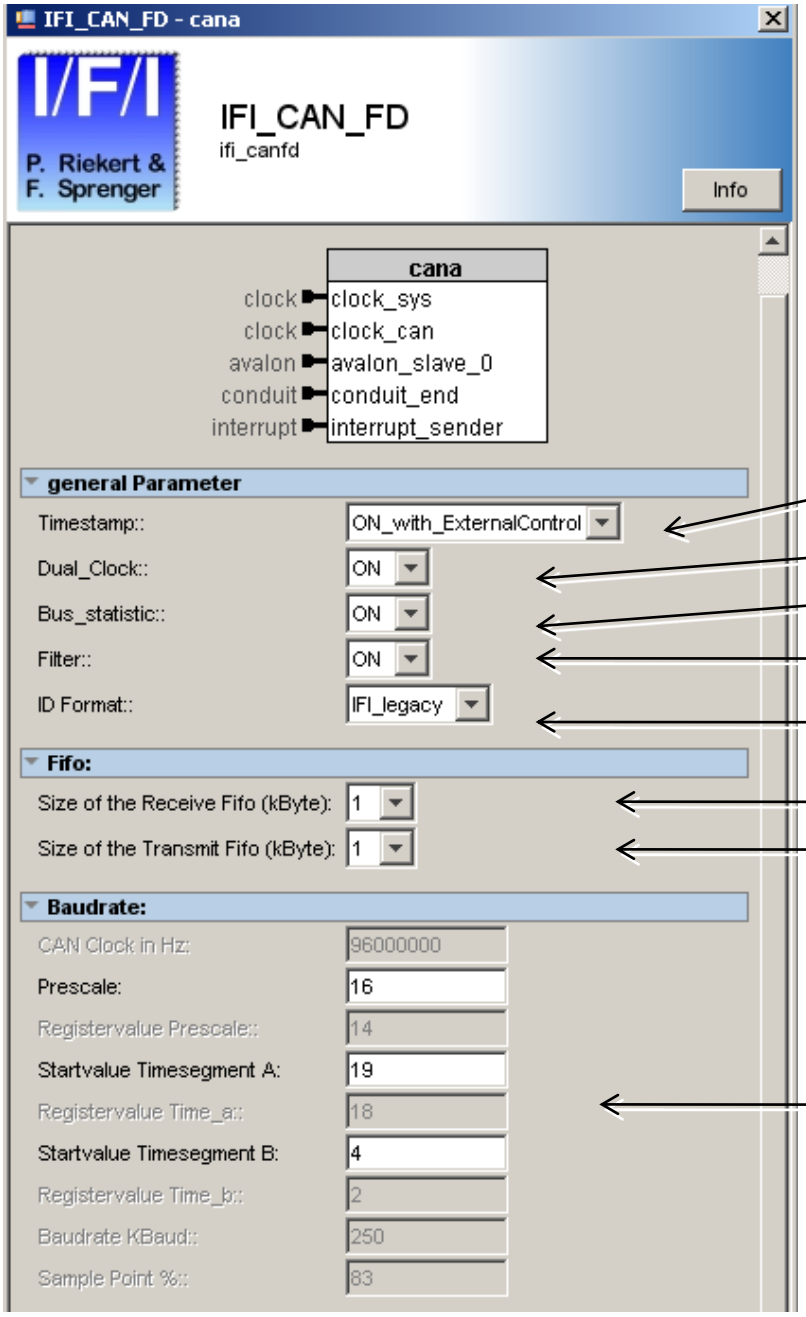


- Info + Documentation
- Parameters
- Information



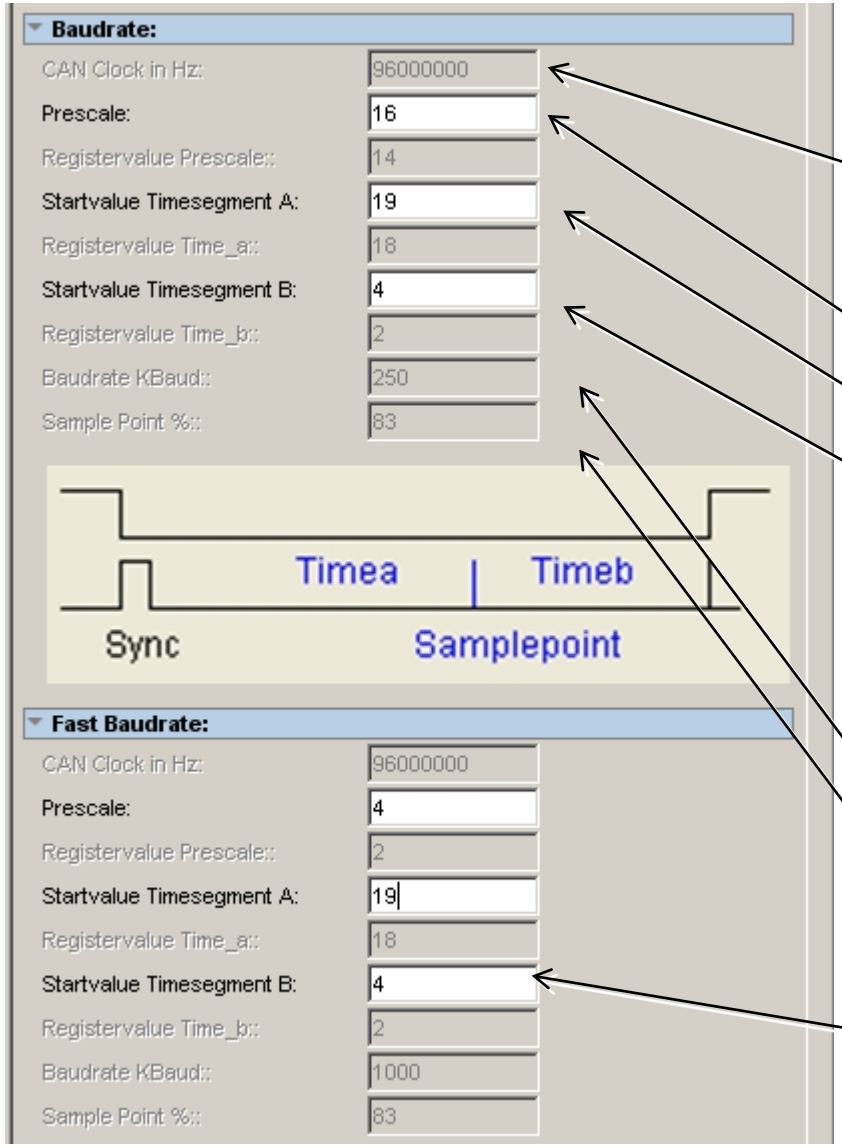


I/F/I F. Sprenger

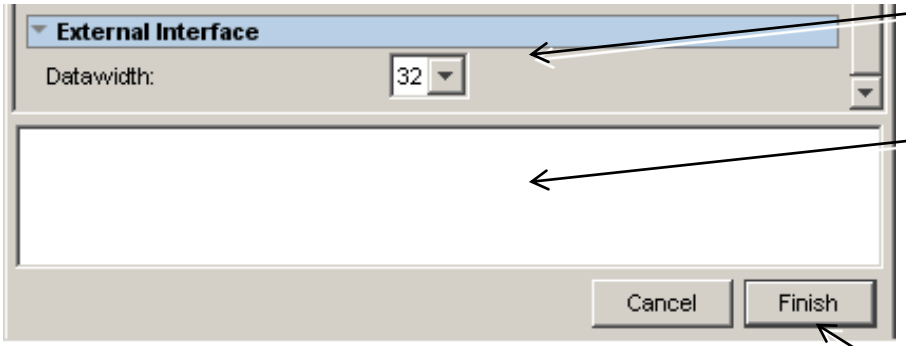


- Enable or Disable the Timestamp
- Enable Dual Clock for System and CAN
- Enable Bus Statistic
- Enable Filters and Masks
- Select ID Format
- Receive Fifo-size in kByte
- Transmit Fifo-size in kByte

- Baudrate Calculator

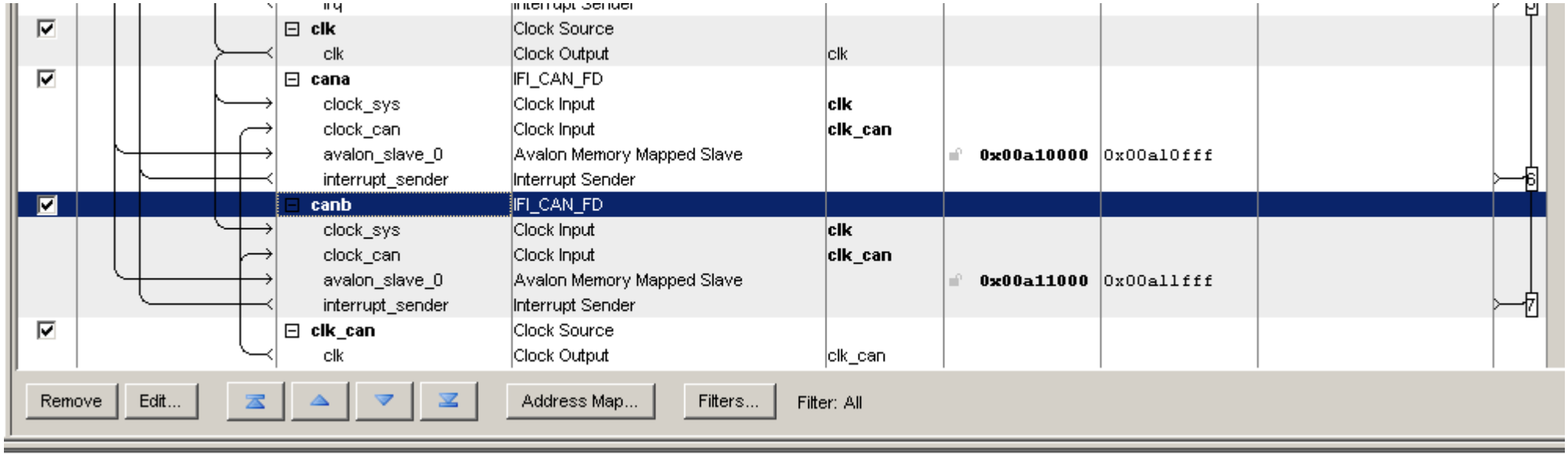


- Both CAN timing parameters are startup values and can be overwritten by software
- The clock frequency of your selected clock (updated after connected in SOPC)
- The CAN timing is always a result of three parameters
  - Timesegment A
  - Timesegment B
  - Prescale
- The necessary registervalues of these parameters are different to the values used for computation of the
- Baudrate and the
- Samplepoint
- Values for the Fast Baudrate (when selected in transmit / receive message)



- For external CPU interfaces it is possible to select the width of the databus.
- 64 Bits / 32Bits / 16 Bits / 8 Bits
- In the information field you will find informations, warnings and errors
- Clicking on Finish will add the core to your SOPC system

Adding the Core to your System



- Specify the desired instance **name**, **base address**, and **IRQ**
- connect the system clock to clock_sys Clock Input
- and connect your can-clock to clock_can when Dual_Clock is enabled

- Complete the system generation as described in the Altera SOPC Builder documentation.

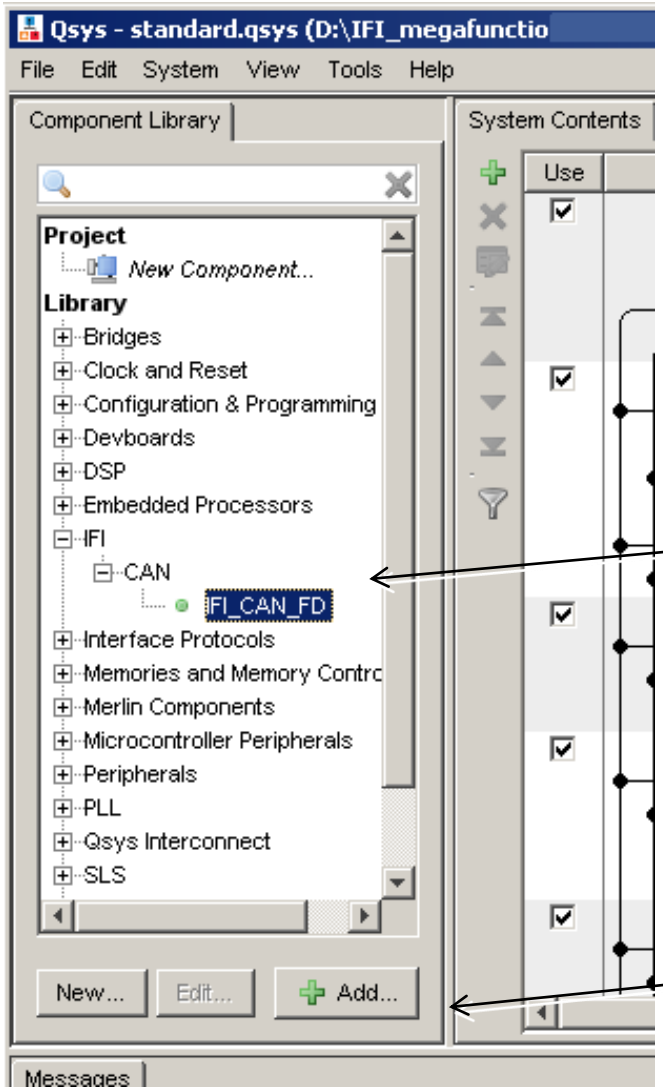
Integrating the Core using QSYS Interconnect / Platform Designer

- *Adding the Core to your QSYS System*
- *About*
- *Documentation*
- *Parameterize*

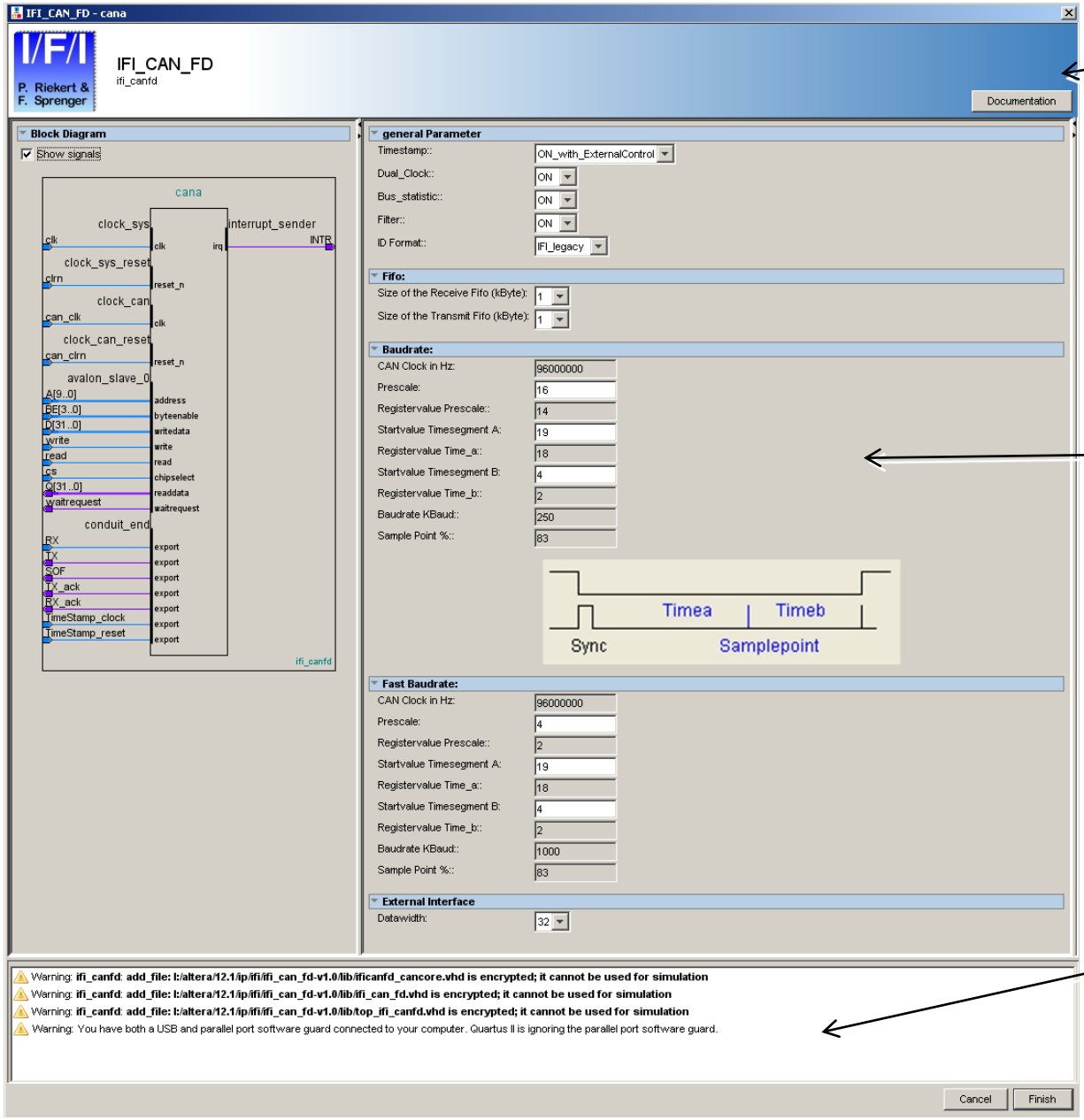
Integrating the Core with your System using QSYS

- This section contains instructions on the following:
  - Adding the core to your system
  - Running the reference design
- These instructions assume that the user is familiar with the
  - Altera OpenCore evaluation process,
  - Altera's Quartus II development software,
  - and the Altera QSYS Interconnect tool
- For more information on these prerequisites, please visit www.altera.com

Adding the Core to your QSYS System

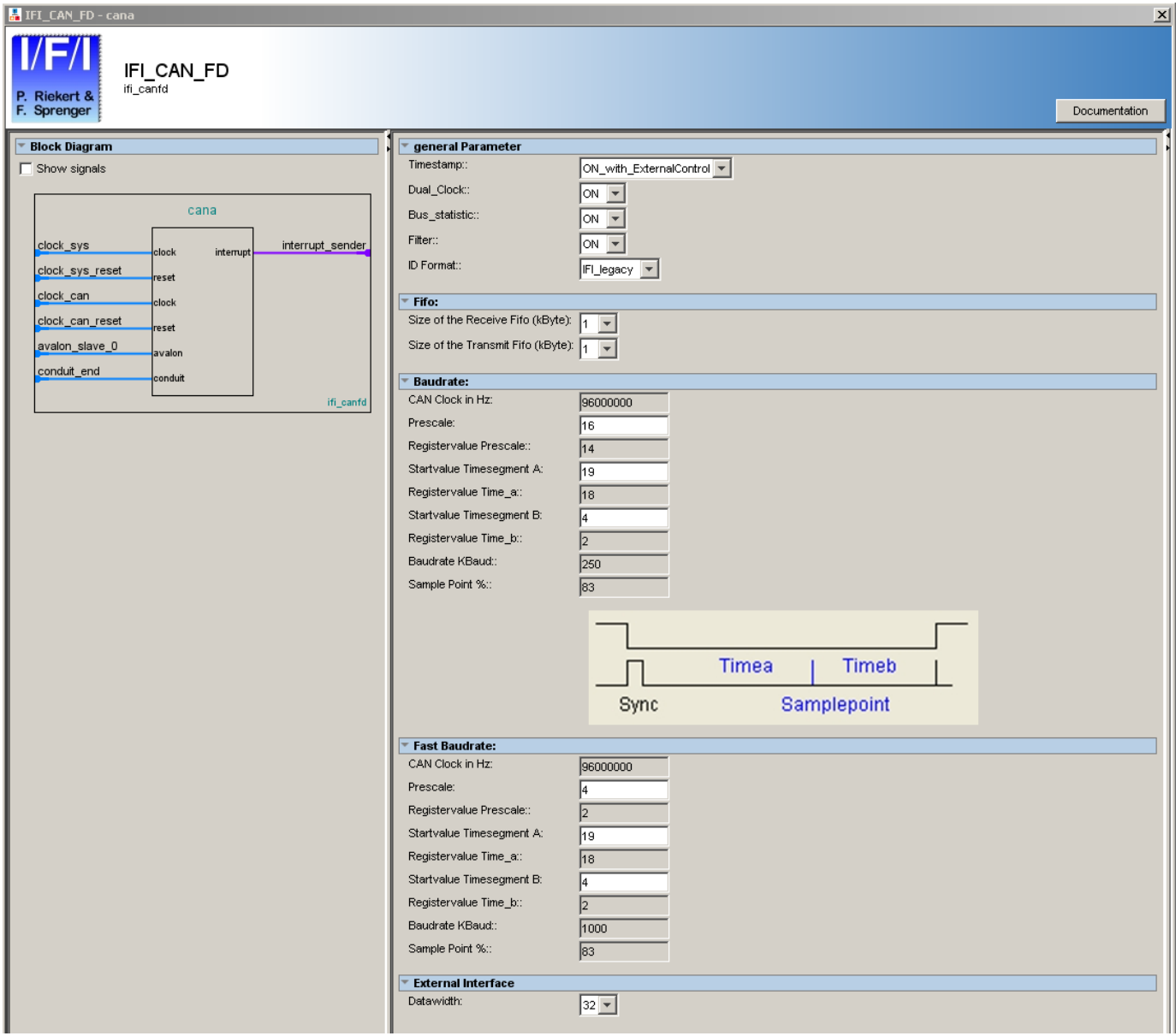


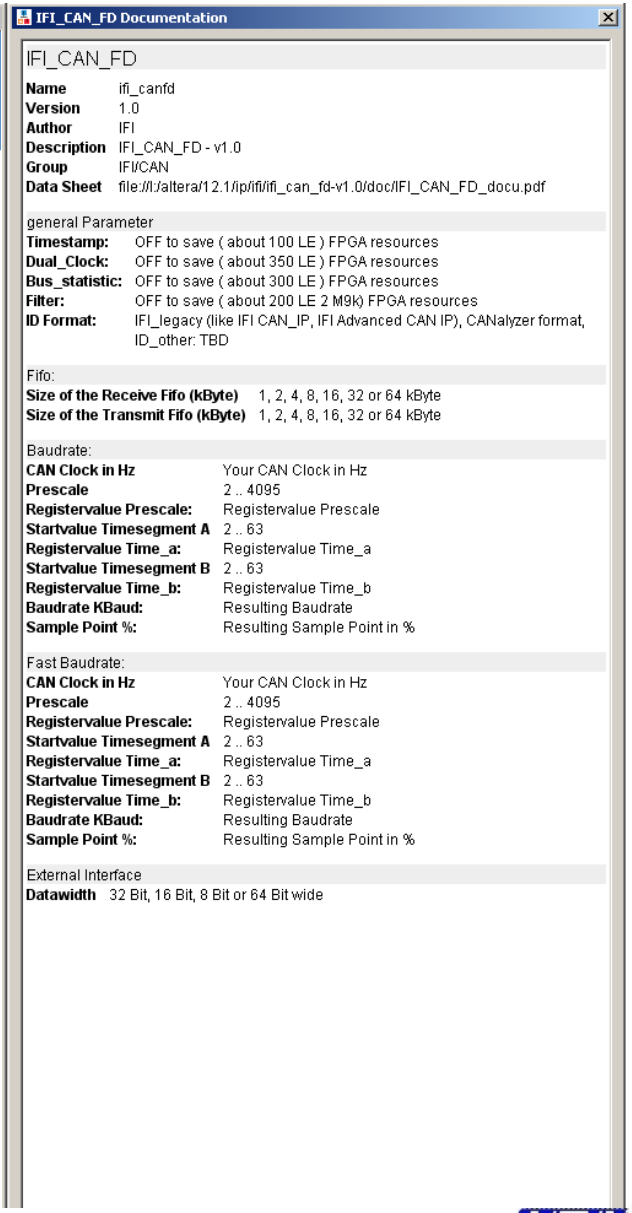
- Launch the QSYS Tool from Quartus II (Tools menu).
- Select the core by clicking on the core name
- Click "Add" to add the core to your system.



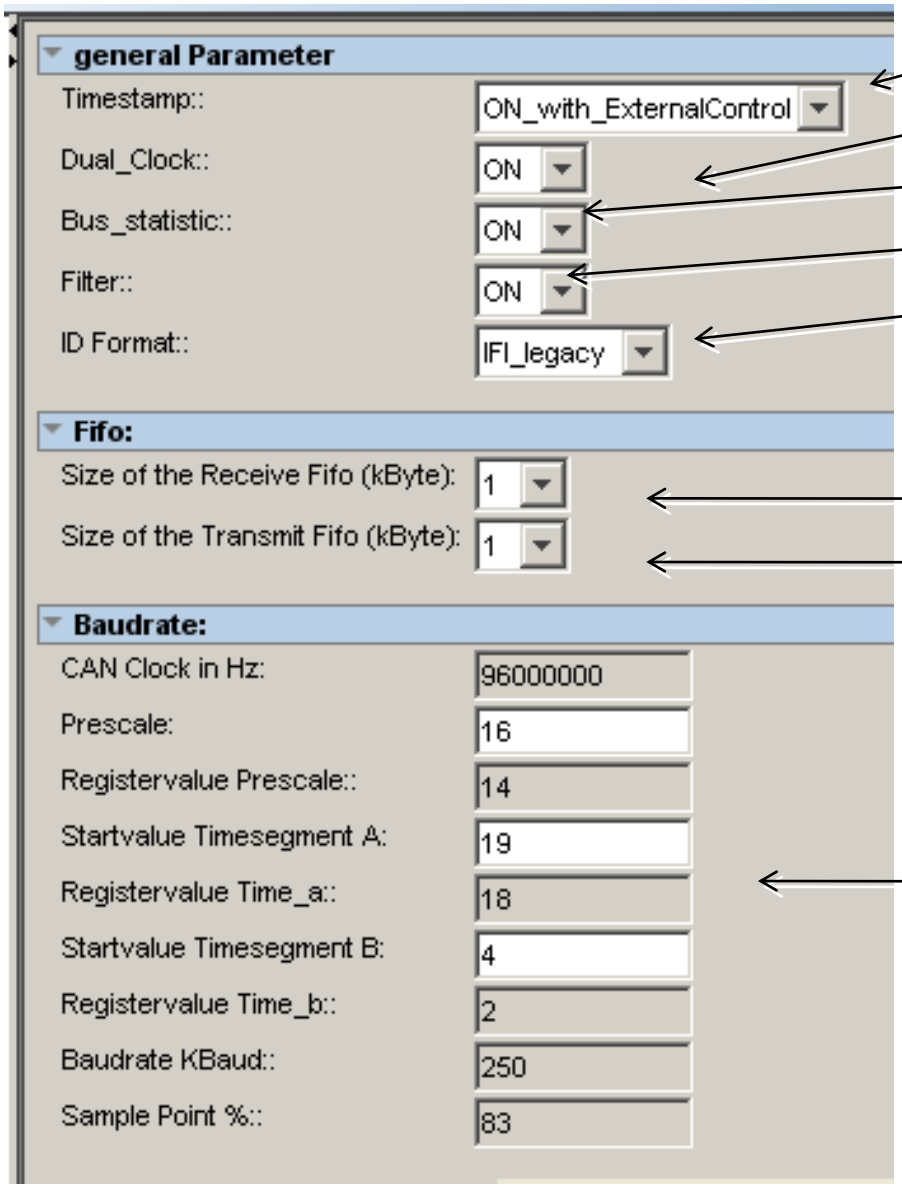
- Info + Documentation
- Parameters
- Information

I/F/I F. Sprenger

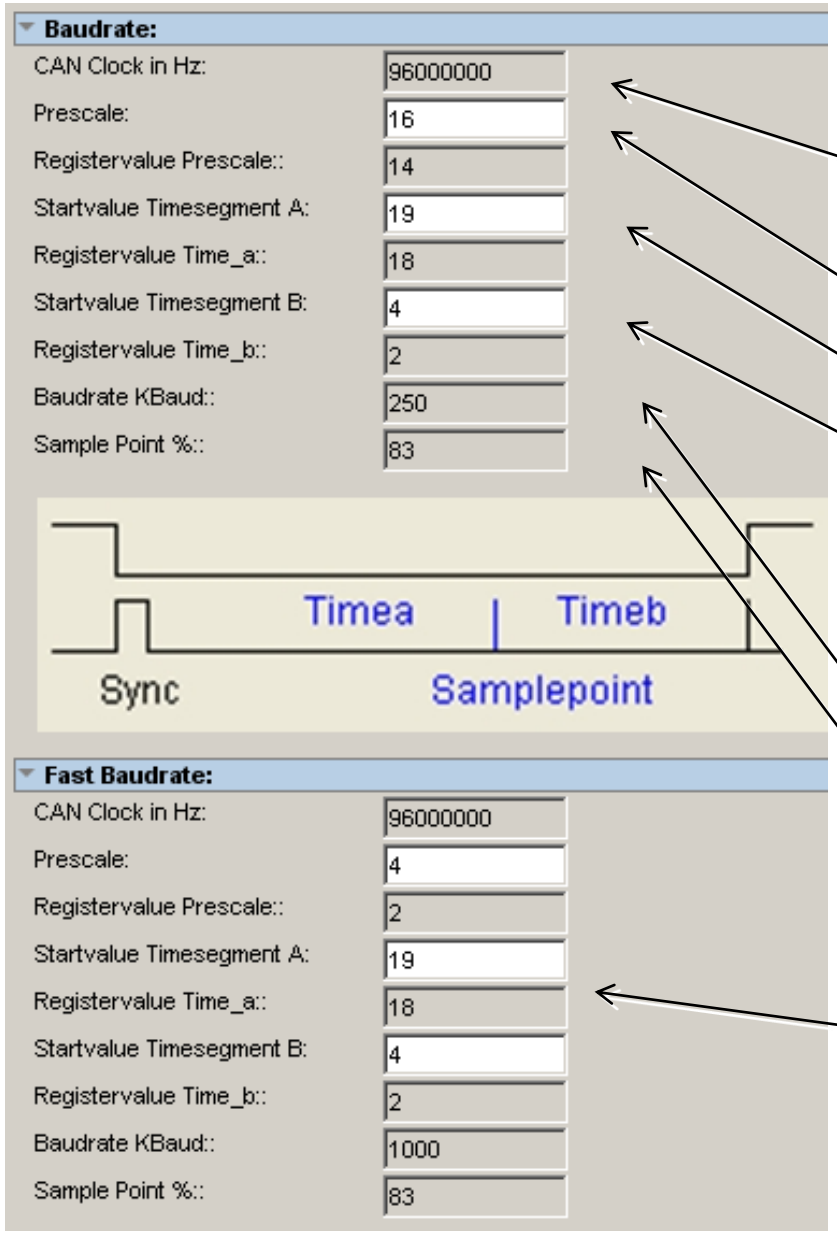




I/F/I F. Sprenger

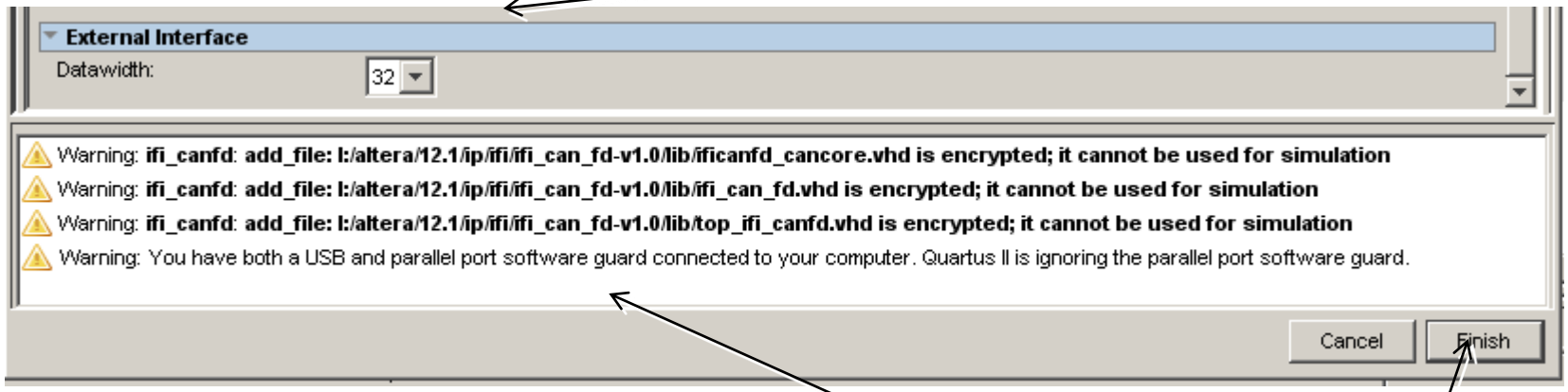


- Enable or Disable the Timestamp
- Enable Dual Clock for System and CAN
- Enable Bus Statistic
- Enable Filters and Masks
- Select ID Format
- Receive Fifo-size in kByte
- Transmit Fifo-size in kByte
- Baudrate Calculator



- Both CAN timing parameters are startup values and can be overwritten by software
- The clock frequency of your selected clock (updated after connected in QSYS)
- The CAN timing is always a result of three parameters
  - Prescale
  - Timesegment A
  - Timesegment B
- The necessary register values of these parameters are different to the values used for computing the
- Baudrate and the
- Samplepoint
- Values for the Fast Baudrate (when selected in transmit / receive message)

- For external CPU interfaces it is possible to select the width of the databus.
- 64 Bits / 32Bits / 16 Bits / 8 Bits



- In the information field you will find informations, warnings and errors
- Clicking on Finish will add the core to your SOPC system

Adding the Core to your System

- Specify the desired instance **name**, **base address**, and **IRQ**
- connect the system clock to clock_sys Clock Input
- and connect your can-clock to clock_can when Dual_Clock is enabled
- and connect your can-reset to clock_can_reset when Dual_Clock is enabled

- Complete the system generation as described in the Altera QSYS documentation

Reference Designs

- *Running a Reference Design*
- *Creating a Software Project*
- *Running a Hardware Configuration*

Running a Reference Design

- Start Quartus II, version 9.1 or a newer version
- Open one of the Quartus II projects in
  - <Core installation directory>\ reference_designs\xxx\IFI_CANFD_Ref_design.qpf
- Launch SOPC Builder from Quartus II (Tools menu)
- or Launch QSYS on newer Quartus II versions
- Click "Generate" to generate the HDL and Modelsim project files
- Click "Exit" to go back to Quartus and compile the design
- Launch the IDE or SBT for the creation of software projects or Modelsim software simulation
  - For the simulation are the following lines in the Testbench included
    rx_to_the_cana <= tx_from_the_cana and tx_from_the_canb;
    rx_to_the_canb <= tx_from_the_cana and tx_from_the_canb;
  - This allows communication between both CAN nodes

Creating a Software Project (SBT)

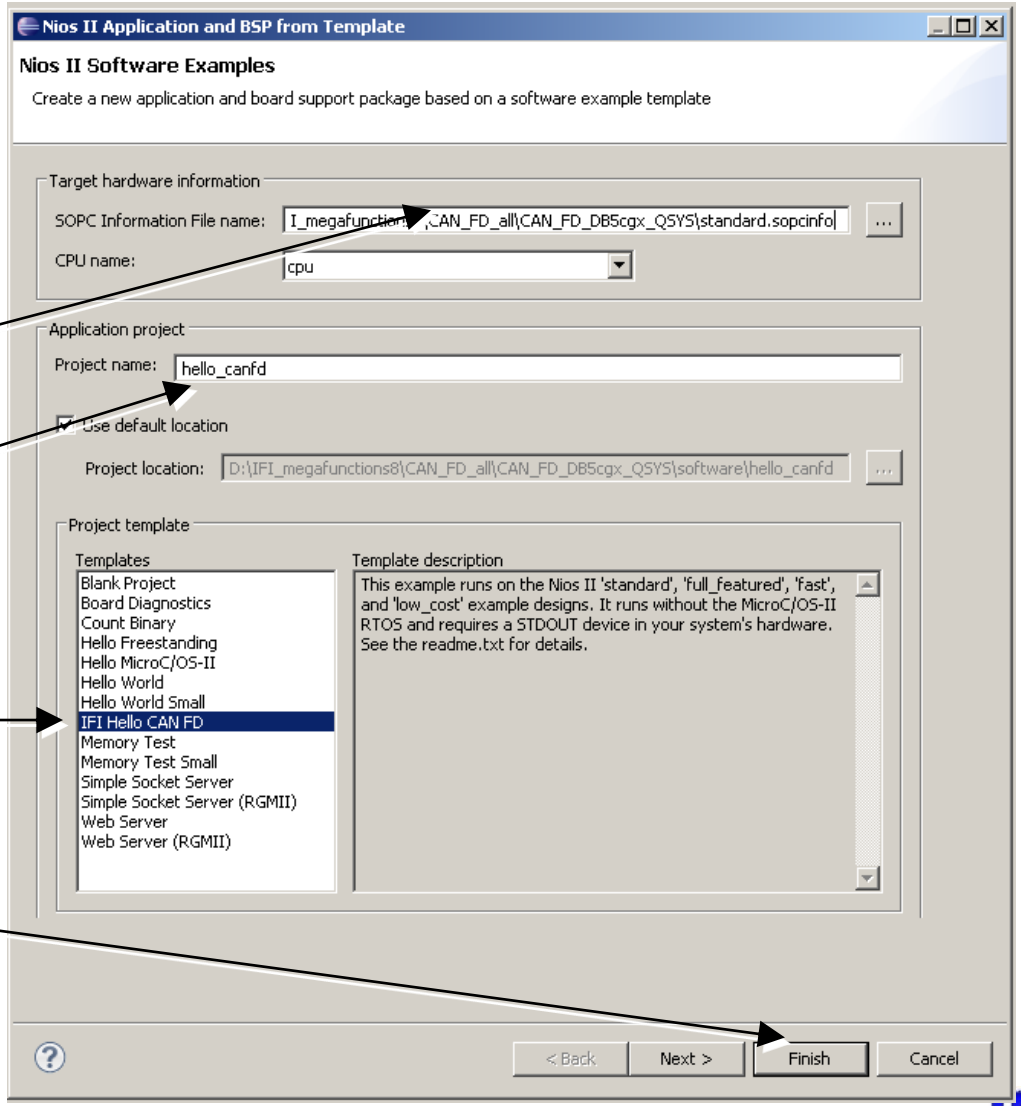
- File → New →
  - NiosII application and BSP from Template

Select the SOPCINFO of the project

Give a project name

Select „IFI Hello CAN FD“

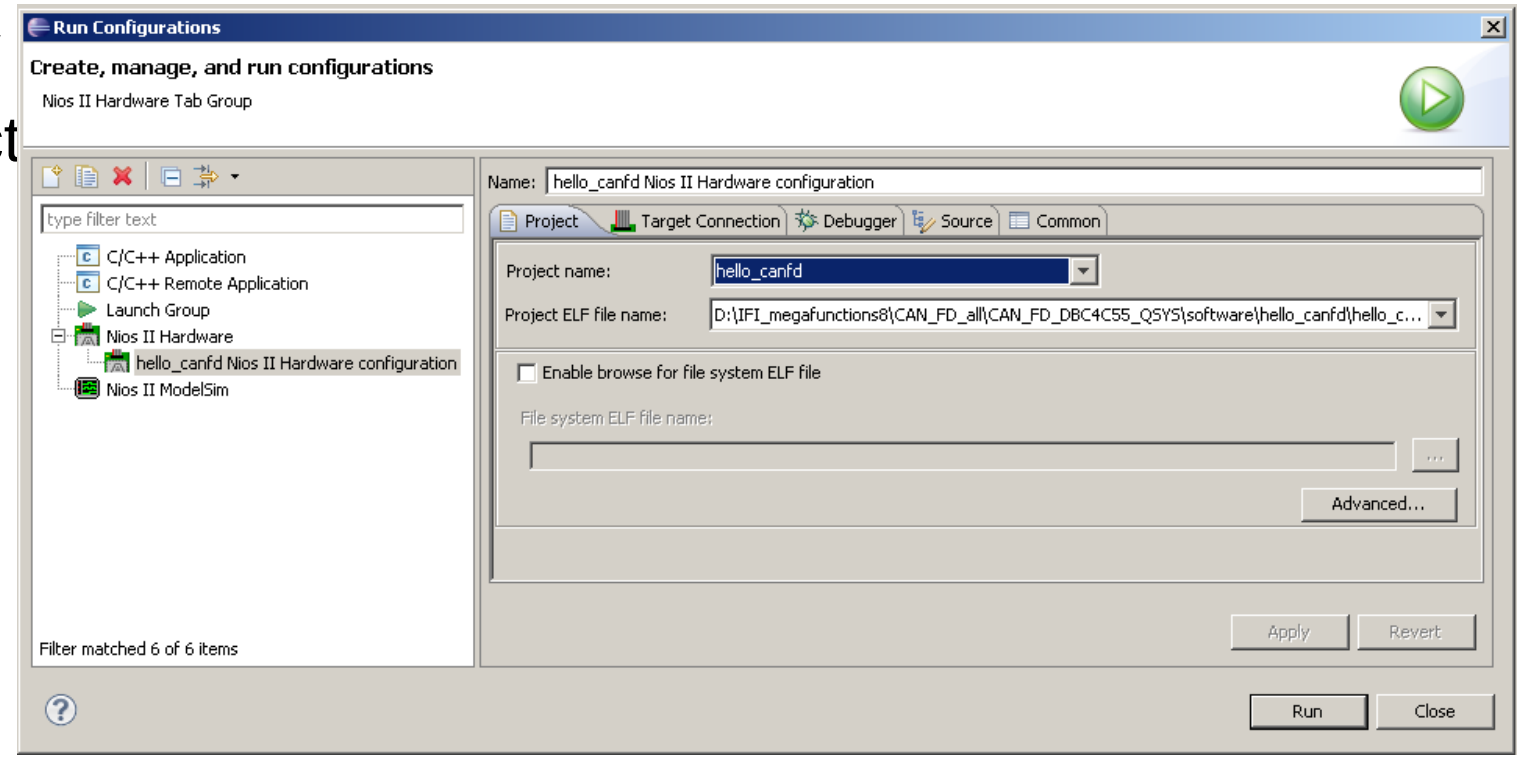
Click on Finish



I/F/I F. Sprenger

Run a Hardware Configuration

- Select your project within the C/C++ Projects View
- Run → Run Configurations ...
- Select NiosII Hardware
- Click on New
- Select Project
- Click on Run

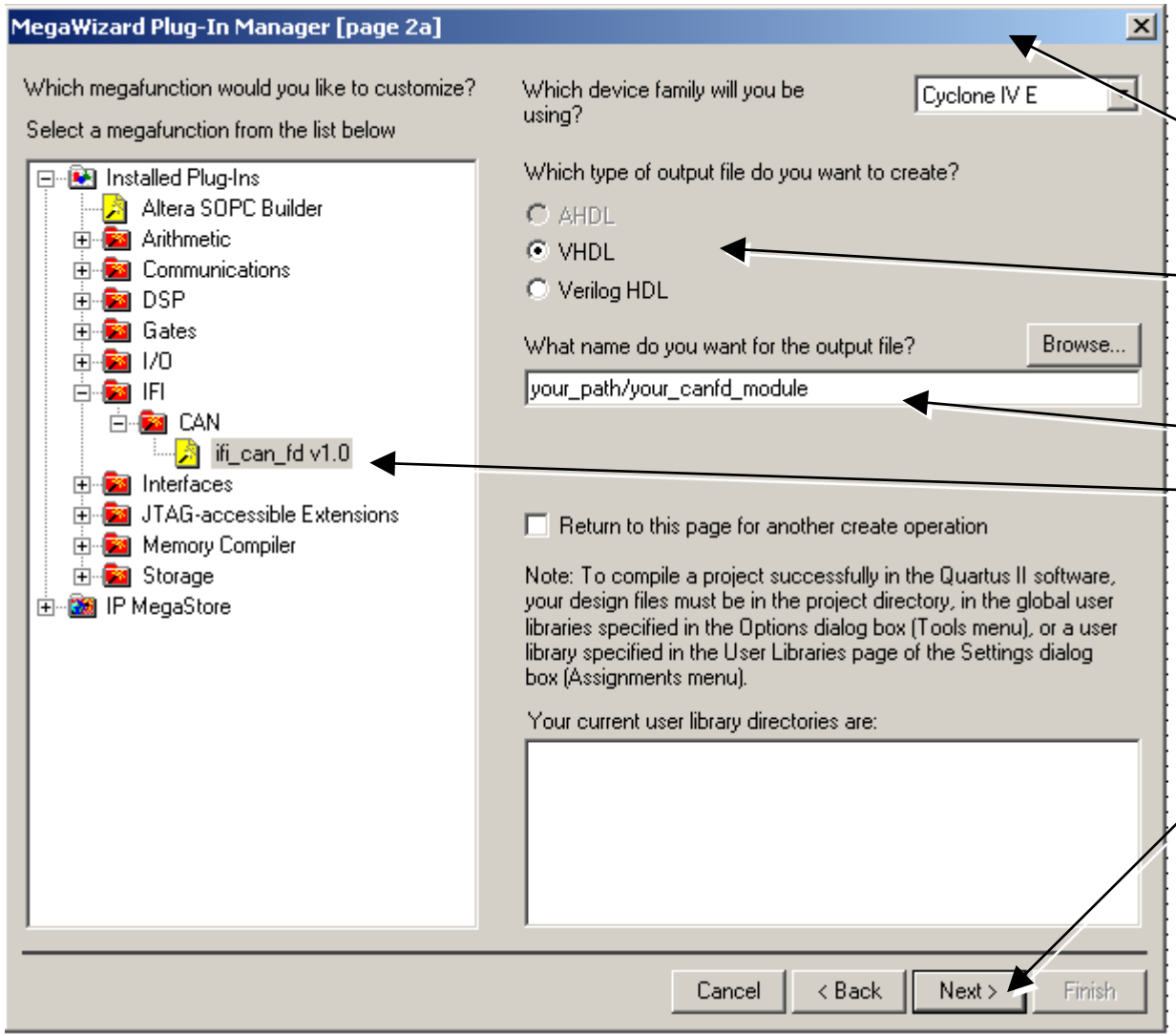


I/F/I
F. Sprenger

Using the Core without Nios/SOPC/QSYS

- *MegaWizard Plug-In Manager*
- *Parameterize*
- *Generate*
- *Quartus Symbol*
- *Reopening of the Modul*
- *Port Description*
- *ReadWrite Timing*

MegaWizard Plug-In Manager



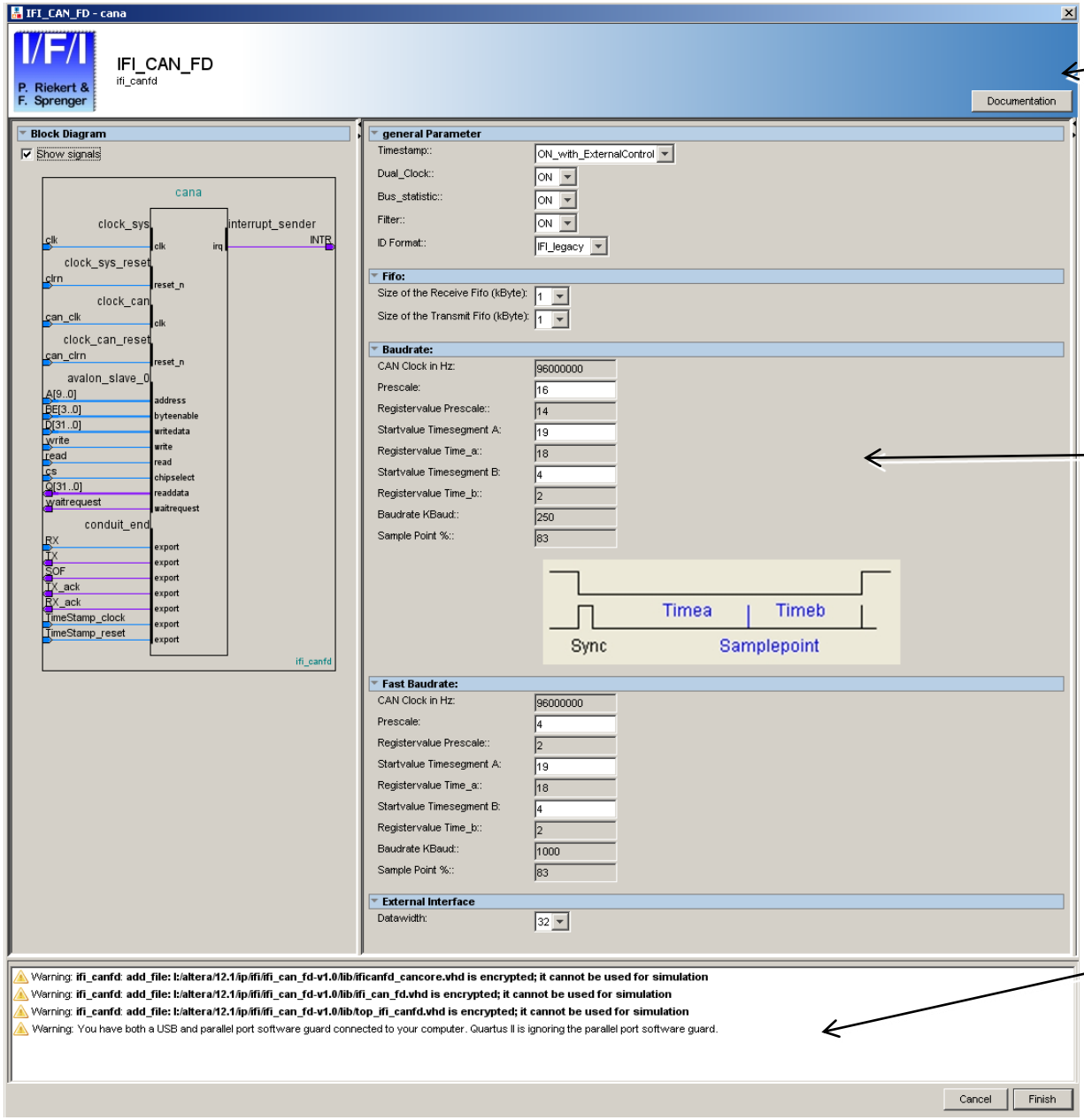
Start MegaWizard Plug-In Manager

Select the wrapper and simulation language

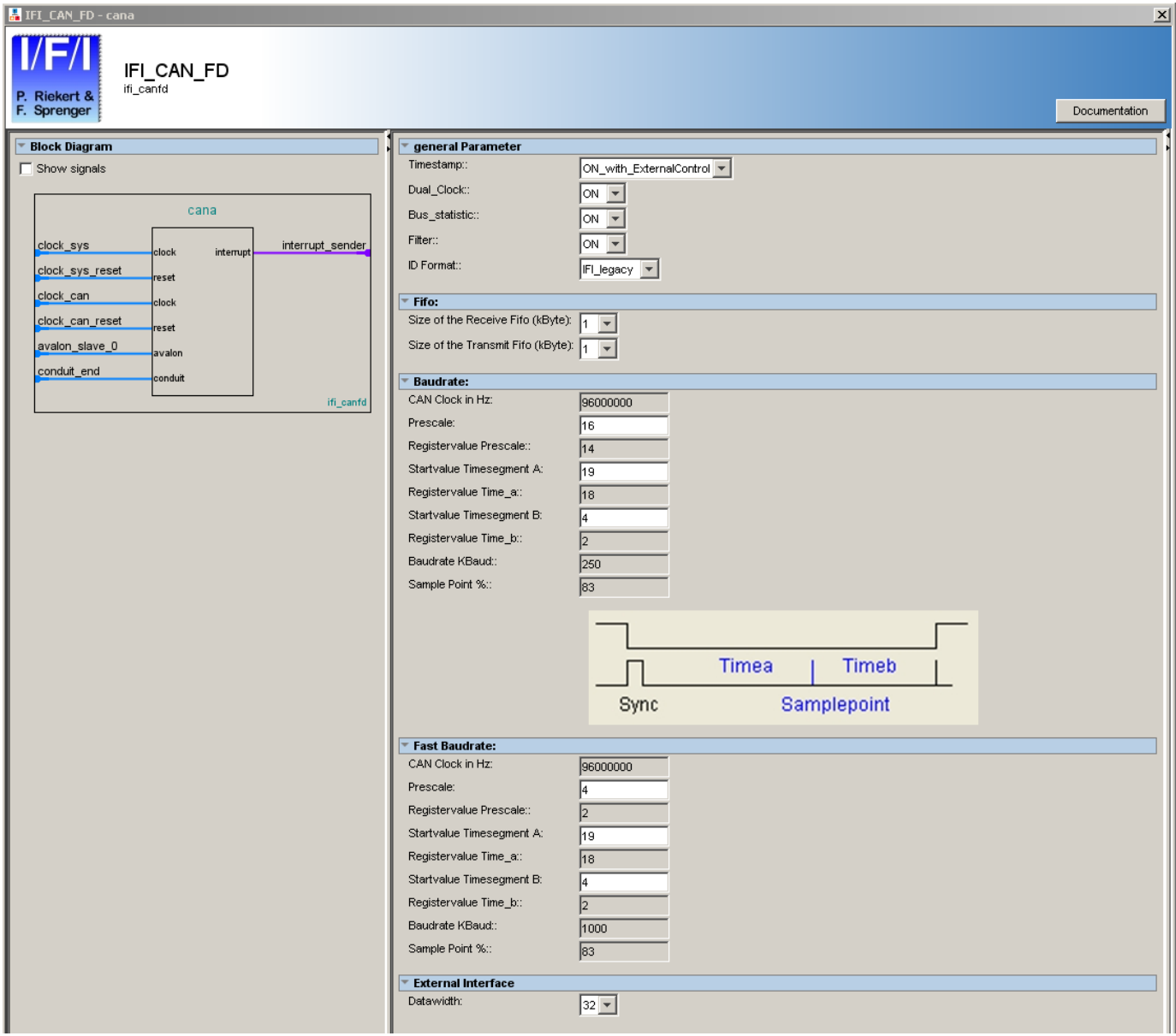
Type in the name

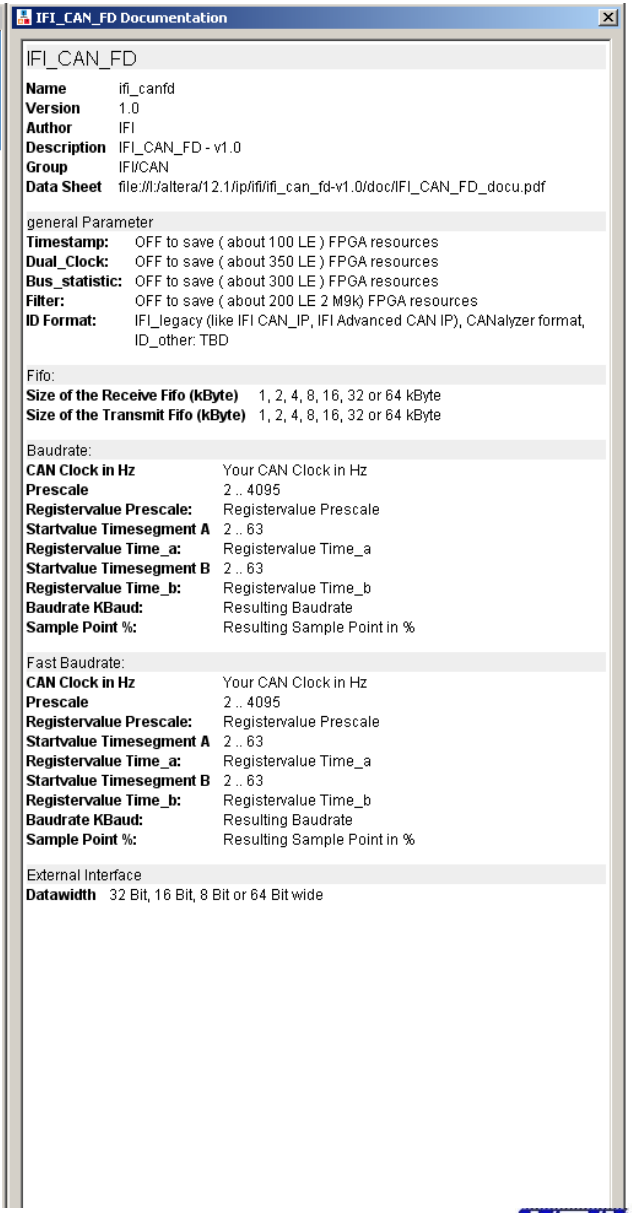
Select the IP Core

Click Next

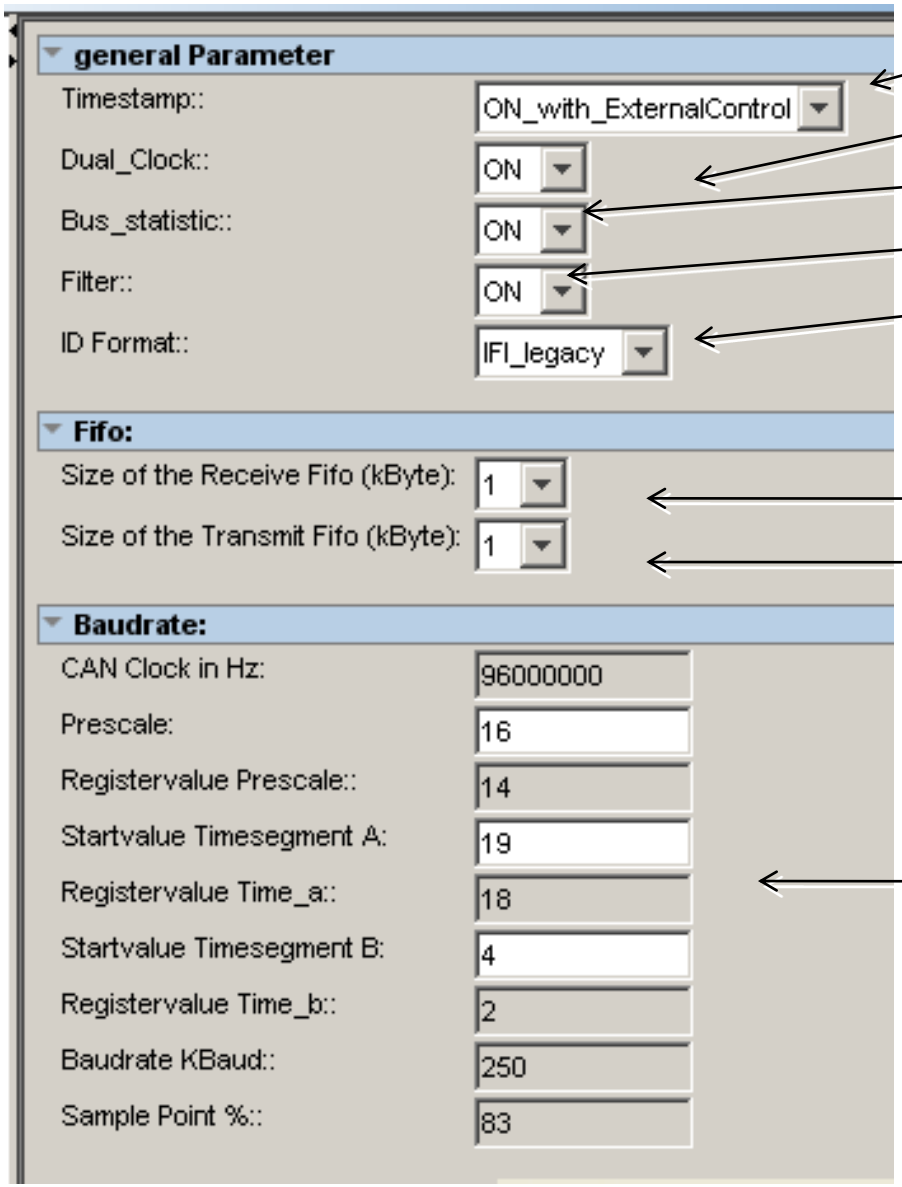


- Info + Documentation
- Parameters
- Information

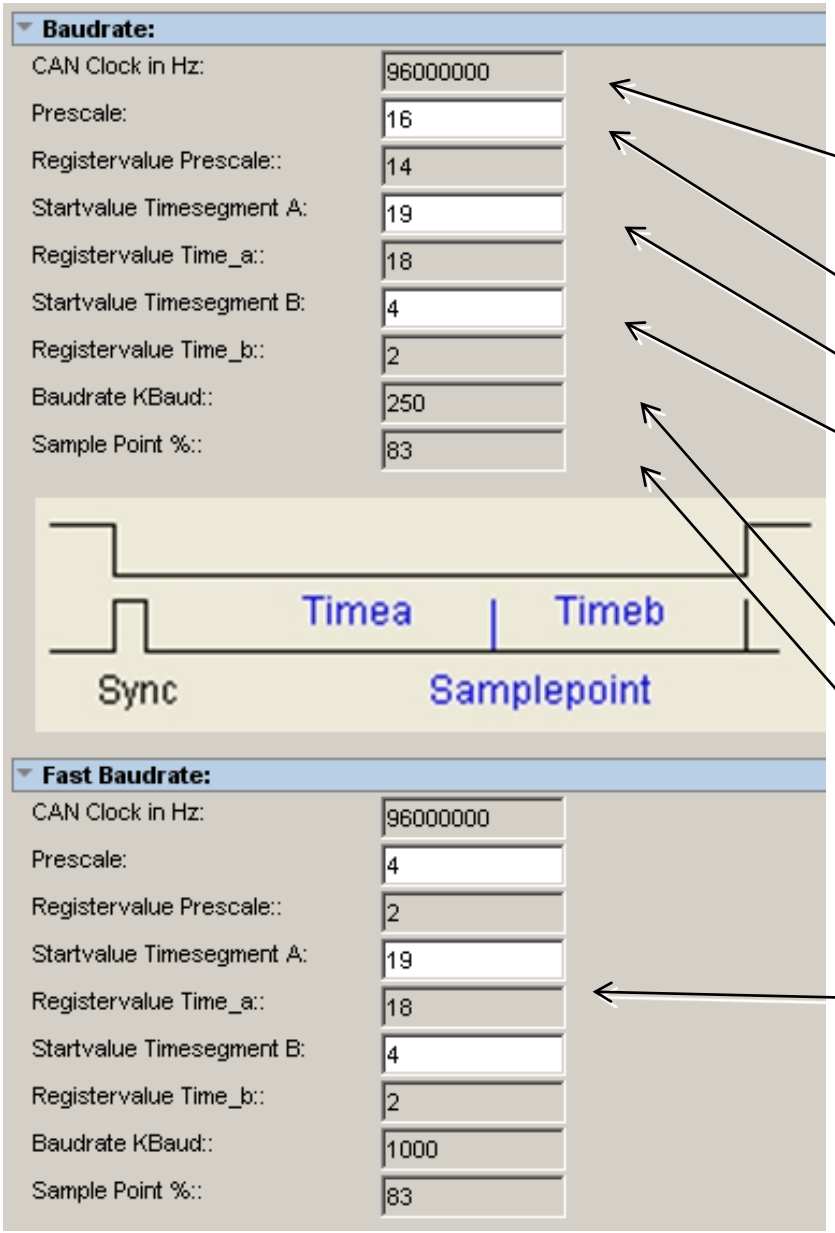




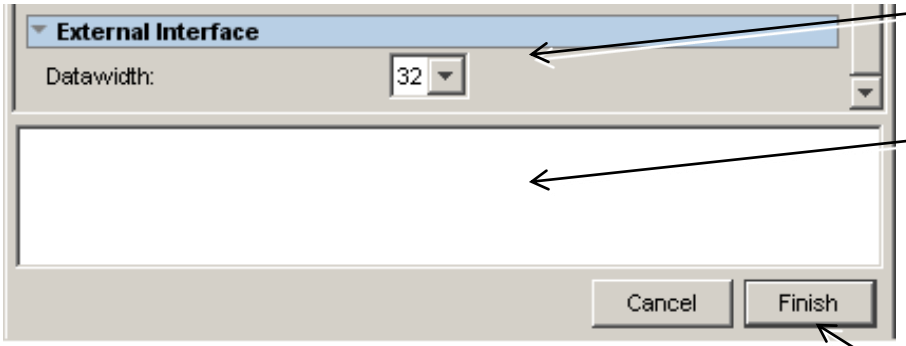
I/F/I F. Sprenger



- Enable or disable the Timestamp
- Enable Dual Clock for system and CAN
- Enable Bus statistic
- Enable filters and masks
- Select ID format
- Receive Fifo-size in kByte
- Transmit Fifo-size in kByte
- Baudrate Calculator



- Both CAN timing parameters are startup values and can be overwritten by software
- The clock frequency of your clock
  - With Megawizzard this is not updated by the IP
  - It is used for this GUI only, so ignore the computation here
- The CAN timing is always a result of three parameters
  - Prescale
  - Timesegment A
  - Timesegment B
- The necessary register values of these parameters are different to the values used for computing the
- Baudrate and the
- Samplepoint
- Values for the Fast Baudrate (when selected in transmit / receive message)

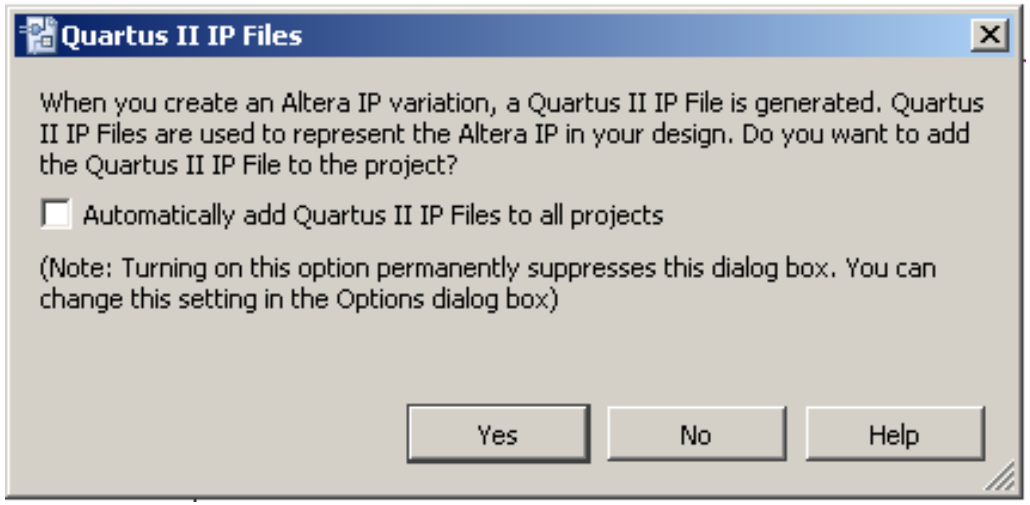


- For external CPU interfaces it is possible to select the width of the databus.
- 64 Bits / 32Bits / 16 Bits / 8 Bits
- In the information field you will find information, warnings and errors
- Click on Finish will add the core to your system

Generation of the Core Variation

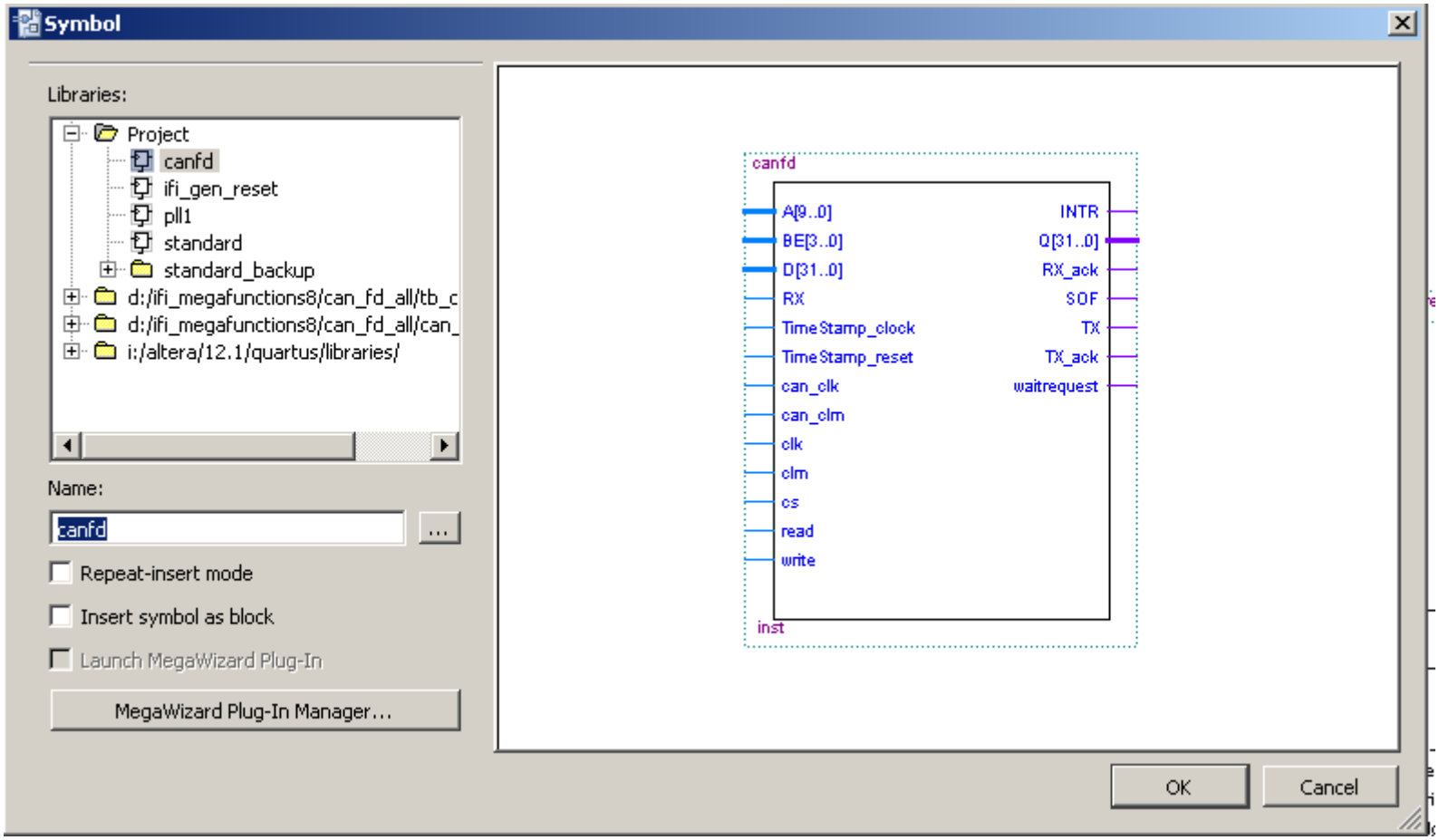
- After successfull generation click on Exit

Use the QuartusII IP File



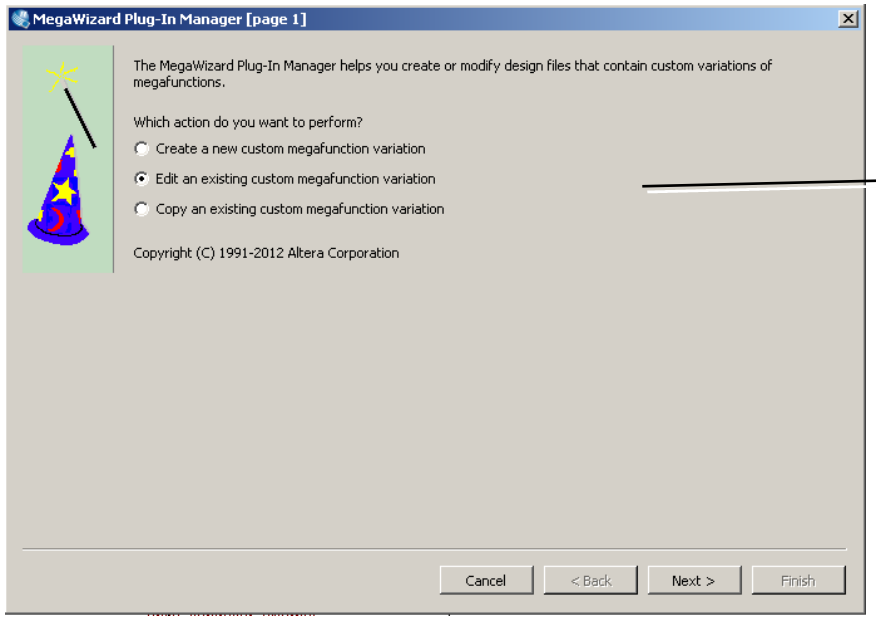
- Click on Yes

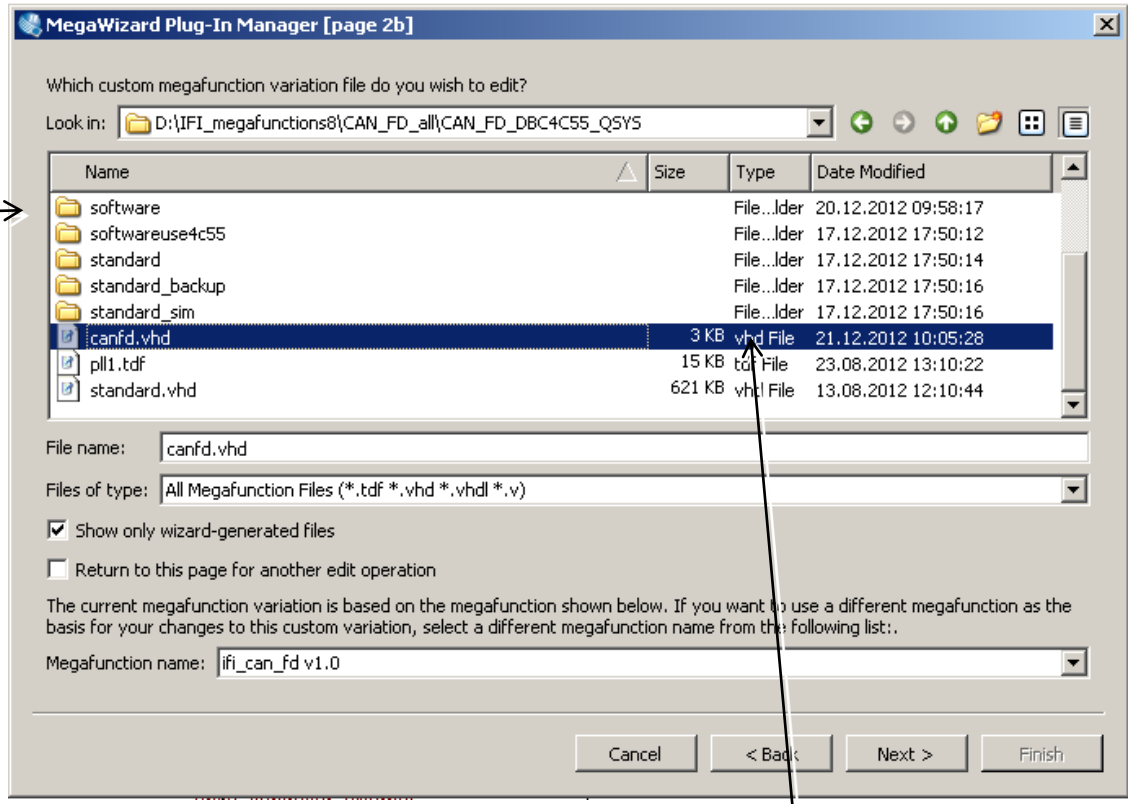
Your new QuartusII IP Symbol



- Click on OK

Re-opening of the Module





- Start the MegaWizard Plug-In manager
- Select Edit
- Select your component and click Next

default Port description

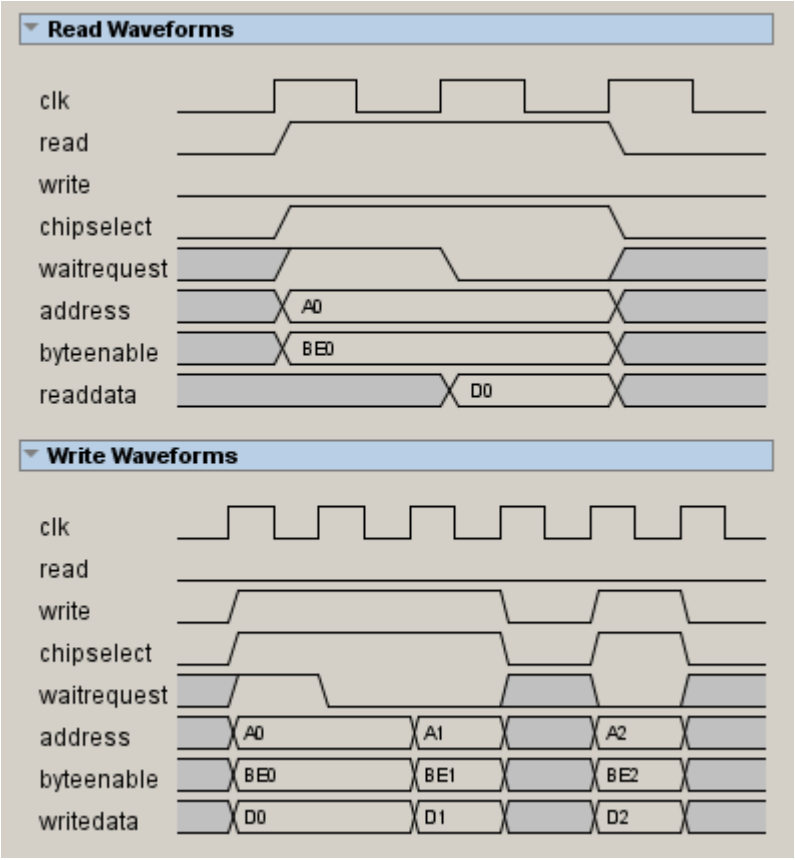
Portname	Direction	Usage	Description
TX	output	external	Transmit data from CAN_FD IP-core
RX	input	external	Receive data in CAN_FD IP-core
clk	input	internal	System clock
clrn	input	internal	System reset (low active)
read	input	internal	Read request
write	input	internal	Write request
cs	input	internal	Chip select
waitrequest	output	internal	Waitrequest (identical to Avalon Memory Mapped Slave)
A[addresswidth-1..0]	input	internal	Address for read/write requests
D[datawidth-1..0]	input	internal	Write data bus
Q[datawidth-1..0]	output	internal	Read data bus
BE[byteenables-1..0]	input	internal	Byteenable
INTR	output	internal	Interrupt request

additional Port description

Portname	Direction	Usage	Description
TimeStamp_clock	input	external	Clock for TimeStamp
TimeStamp_reset	input	external	Reset for TimeStamp (high active)
SOF	output	external	Pulse when SOF (Start Of Frame) is detected
RX_ack	output	external	Pulse when a received frame was accepted
TX_ack	output	external	Puls when a transmitted frame was accepted

- Select TimeStamp_ExternalControl to get these additional inputs and outputs for your own TimeStamp design

ReadWrite Timing



- The MegaWizard implementation
  - use the same timing as with the avalon memory mapped slave

- Details: www.altera.com/literature/manual/mnl_avalon_spec.pdf

License Agreement

License Agreement

PLEASE REVIEW THE FOLLOWING TERMS AND CONDITIONS CAREFULLY BEFORE USING THE IFI IP-MODULE. BY USING THE IFI IP-MODULE AND/OR PAYING A LICENSE FEE, YOU INDICATE YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS, WHICH CONSTITUTE THE LICENSE AGREEMENT (the "AGREEMENT") BETWEEN YOU AND IFI. IN THE EVENT THAT YOU DO NOT AGREE WITH ANY OF THESE TERMS AND CONDITIONS, DO NOT USE THE IFI IP-MODULE AND WE ASK YOU TO DESTROY ANY COPIES YOU HAVE MADE IMMEDIATELY.

DEFINITIONS:

"Party" means either IFI or YOU.

"Specification" means IFI's technical description for the IFI IP-MODULE covered by this Agreement to the extent such technical description relates to the operation, performance and other material attributes of the IFI IP-MODULE.

1. License to the IFI IP-MODULE:

1.1 Subject to the terms and conditions of this Agreement (including but not limited to YOUR payment of the license fee set forth in Paragraph 4.0), IFI grants to YOU a single-user, non-transferable, non-exclusive and (except as specified by IFI) perpetual license to use the IFI IP-MODULE as follows. YOU may:

(a) design with, parameterize, compile and route the IFI IP-MODULE;
(b) program Altera devices with the IFI IP-MODULE;
(c) use the IFI IP-MODULE on a single computer only; and
(d) except as otherwise provided in Paragraph 1.2, YOU may use, distribute, sell and/or otherwise market products containing licensed products to any third party in perpetuity. YOU may also sublicense YOUR right to use and distribute products containing licensed products as necessary to permit YOUR distributors to distribute and YOUR customers to use products containing licensed products. YOU are expressly prohibited from using the IFI IP-MODULE to design, develop or program Non-Altera Devices .

1.2 YOU may make only one copy of the IFI IP-MODULE for back-up purposes only. The IFI IP-MODULE may not be copied to, installed on or used with any other computer, or accessed or otherwise used over any network, without prior written approval from IFI.

1.3 Any copies of the IFI IP-MODULE made by or for YOU shall include all intellectual property notices, including copyright and proprietary rights notices, appearing on such IFI IP-MODULE. Any copy or portion of the IFI IP-MODULE, including any portion merged into a design and any design or product that incorporates any portion of the IFI IP-MODULE, will continue to be subject to the terms and conditions of this Agreement.

1.4 The source code of the IFI IP-MODULE, and algorithms, concepts, techniques, methods and processes embodied therein, constitute trade secrets and confidential and proprietary information of IFI and its licensors and LICENSEE shall not access or use such trade secrets and information in any manner, except to the extent expressly permitted herein. IFI and its licensors retain all rights with respect to the IFI IP-MODULE, including any copyright, patent, trade secret and other proprietary rights, not expressly granted herein.

2. License Restrictions:

YOU MAY NOT USE THE IFI IP-MODULE EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS AGREEMENT OR SUBLICENSE OR TRANSFER THE IFI IP-MODULE OR RIGHTS WITH RESPECT THERETO. YOU MAY NOT DECOMPILE, DISASSEMBLE, OR OTHERWISE REVERSE ENGINEER THE IFI IP-MODULE OR ATTEMPT TO ACCESS OR DERIVE THE SOURCE CODE OF THE IFI IP-MODULE OR ANY ALGORITHMS, CONCEPTS, TECHNIQUES, METHODS OR PROCESSES EMBODIED THEREIN; PROVIDED, HOWEVER, THAT IF YOU ARE LOCATED IN A MEMBER NATION OF THE EUROPEAN UNION OR OTHER NATION THAT PERMITS LIMITED REVERSE ENGINEERING NOTWITHSTANDING A CONTRACTUAL PROHIBITION TO THE CONTRARY, YOU MAY PERFORM LIMITED REVERSE ENGINEERING, BUT ONLY AFTER GIVING NOTICE TO IFI AND ONLY TO THE EXTENT PERMITTED BY THE APPLICABLE LAW IMPLEMENTING THE EU SOFTWARE DIRECTIVE OR OTHER APPLICABLE LAW NOTWITHSTANDING A CONTRACTUAL PROHIBITION TO THE CONTRARY.

3. Term:

This Agreement is effective until terminated. YOU may terminate it at any time by destroying the IFI IP-MODULE together with all copies and portions thereof in any form (except as provided below). It will also terminate immediately if YOU breach any term of this Agreement and upon conditions set forth elsewhere in this Agreement. Upon any termination of this Agreement, YOU shall destroy the IFI IP-MODULE, including all copies and portions thereof in any form (whether or not merged into a design or Licensed Product), and YOUR license and rights under this Agreement shall terminate except that YOU and YOUR customers may continue to sell and use Licensed Products which have been developed in accordance with this Agreement and shipped prior to the termination. In no event may any portions of the IFI IP-MODULE be used in development after termination. In the event of termination for any reason, the rights, obligations and restrictions under Paragraphs 2, 4, 9, and 10 shall survive termination of this Agreement.

4. Payment:

In consideration of the license granted by IFI under Paragraph 1.1 and other rights granted under this Agreement, YOU shall pay the license fee for the IFI IP-MODULE that has been specified by IFI. Such payment shall, as directed by IFI, be made directly to IFI .YOU shall pay all taxes and duties associated with this Agreement, other than taxes based on IFI's income.

5. Maintenance and Support:

IFI shall, but only until the date, in the format YYYY.MM, provided in the license file for a IFI IP-MODULE ("Maintenance Expiration Date"):

5.1 use commercially reasonable efforts to provide YOU with fixes to defects in the IFI IP-MODULE that cause the IFI IP-MODULE not to conform substantially to the Specifications and that are diagnosed as such and replicated by IFI;

5.2 provide YOU with fixes and other updates to the IFI IP-MODULE that IFI chooses to make generally available to its customers without a separate charge; and

5.3 respond by telephone or email to inquiries from YOU.

6. Limited Warranties and Remedies:

6.1 IFI represents and warrants that, until the Maintenance Expiration Date ("Warranty Period"), the IFI IP-MODULE will substantially conform to the Specifications. YOUR sole remedy, and IFI's sole obligation, for a breach of this warranty shall be (a) for IFI to use commercially reasonable efforts to remedy the non-conformance or (b) if IFI is unable substantially to remedy the non-conformance, for YOU to receive a refund of license fees paid during the previous one (1) year for the defective IFI IP-MODULE. If YOU receive such a refund, YOU agree that YOUR license and rights under this Agreement for the defective IFI IP-MODULE shall immediately terminate and YOU agree to destroy the defective IFI IP-MODULE, including all copies thereof in any form and any portions thereof merged into a design or product, and to certify the same to IFI.

6.2 The foregoing warranties apply only to IFI IP-MODULEs delivered by IFI. The warranties are provided only to YOU, and may not be transferred or extended to any third party, and apply only during the Warranty Period for claims of breach reported (together with evidence thereof) during the Warranty Period. YOU shall provide IFI with such evidence of alleged non-conformities or defects as IFI may request, and IFI shall have no obligation to remedy any non-conformance or defect it cannot replicate. The warranties do not extend to any IFI IP-MODULE which have been modified by anyone other than IFI.

7. Representation:

Each party represents that it has the right to enter into this Agreement and to perform its obligations hereunder.

8. Indemnification:

8.1 Expressly subject to Section 9, IFI shall defend YOU against any proceeding brought by a third party to the extent based on a claim that the IFI IP-MODULE, as delivered by IFI and as used in accordance with this Agreement, infringes a third party's copyright, trade secret, patent, or any other intellectual property right ("IP right"), and pay any damages awarded in the proceeding as a result of the claim (or pay any amount agreed to by IFI as part of a settlement of the claim), provided that IFI shall have no liability hereunder unless YOU notify IFI promptly in writing of any such proceeding or claim, give IFI sole and complete authority to control the defence and settlement of the proceeding or claim, and provide IFI with any information, materials, and other assistance requested by IFI.

8.2 In the event of any such claim or proceeding or threat thereof, IFI may (and, in the event any such claim or proceeding results in the issuance of an injunction by a court of competent jurisdiction prohibiting YOU from using the IFI IP-MODULE, IFI shall), at its option and expense and subject to the limitations of Paragraph 9, seek a license to permit the continued use of the affected IFI IP-MODULE or use commercially reasonable efforts to replace or modify the IFI IP-MODULE so that the replacement or modified version is non-infringing or has a reduced likelihood of infringement, provided that the replacement or modified version has functionality comparable to that of the original. If IFI is unable reasonably to obtain such license or provide such replacement or modification, IFI may terminate YOUR license and rights with respect to the affected IFI IP-MODULE, in which event YOU shall return to IFI the affected IFI IP-MODULE, including all copies and portions thereof in any form (including any portions thereof merged into a design or product), and certify the same to IFI, and IFI shall refund the license fee paid by YOU for the affected IFI IP-MODULE.

IFI shall have no liability or obligation to YOU hereunder for any infringement or claim based on or resulting from (a) the combination or use of the IFI IP-MODULE with other products or components; (b) modification of the IFI IP-MODULE by anyone other than IFI, (c) the use of other than the most recent version of the IFI IP-MODULE if the infringement or claim would have been avoided (or the likelihood thereof reduced) by use of the most recent version; (d) requirements specified by YOU; (e) use of the IFI IP-MODULE in any way not contemplated under this Agreement; or (f) any use of the IFI IP-MODULE, to the extent that IFI has indicated in the applicable Specification that third-party licenses 8.3a may be required to use such IFI IP-MODULE if LICENSEE has not obtained the necessary third-party licenses.

8.3a The license does not include the CAN-Network license (Bosch).

8.4 The provisions of this Paragraph 8 state the entire liability and obligations of IFI, and YOUR sole and exclusive rights and remedies, with respect to any proceeding or claim relating to infringement of copyright, trade secret, patent, or any other intellectual property right.

LIMITATIONS OF LIABILITY

9.1 In no event shall the aggregate liability of IFI relating to this Agreement or the subject matter hereof under any legal theory (whether in tort, contract or otherwise), including any liability under Paragraph 8 or for any loss or damages directly or indirectly suffered by YOU relating to the IFI IP-MODULE, exceed the aggregate amount of the license fees paid by YOU in the previous one (1) year under this Agreement.

9.2 IN NO EVENT SHALL IFI BE LIABLE UNDER ANY LEGAL THEORY, WHETHER IN TORT, CONTRACT OR OTHERWISE (a) FOR ANY LOST PROFITS, LOST REVENUE OR LOST BUSINESS, (b) FOR ANY LOSS OF OR DAMAGES TO OTHER SOFTWARE OR DATA, OR (c) FOR ANY INCIDENTAL, INDIRECT, CONSEQUENTIAL OR SPECIAL DAMAGES RELATING TO THIS AGREEMENT OR THE SUBJECT MATTER HEREOF, INCLUDING BUT NOT LIMITED TO THE DELIVERY, USE, SUPPORT, OPERATION OR FAILURE OF THE MEGACORE LOGIC IFI IP-MODULE, EVEN IF IFI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LIABILITY.

10. General:

10.1 YOU may not sublicense, assign, or transfer this license, or disclose any trade secrets embodied in the IFI IP-MODULE, except as expressly provided in this Agreement. Any attempt to sublicense, assign, or otherwise transfer without prior written approval of the other party any of the rights, duties, or obligations hereunder is void.

10.2 This Agreement is entered into for the benefit of IFI and its licensors and all rights granted to YOU and all obligations owed to IFI shall be enforceable by IFI.

10.3 If YOU have any questions concerning this Agreement, including software maintenance or warranty service, YOU should contact IFI Ing.Büro Für Ic-Technologie, Franz Sprenger, Kleiner Weg 3, 97877 Wertheim, Germany.

10.4 YOU agree that the validity and construction of this Agreement and performance hereunder, shall be governed by the laws of German jurisdictions, without reference to conflicts of law principles. YOU agree to submit to the exclusive jurisdiction of the courts in Germany, for the resolution of any dispute or claim arising out of or relating to this Agreement. The Parties hereby agree that the Party who does not prevail with respect to any dispute, claim, or controversy relating to this Agreement shall pay the costs actually incurred by the prevailing Party, including any attorneys' fees.

10.5 No amendment to this Agreement shall be effective unless it is in writing signed by a duly authorized representative of both Parties. The waiver of any breach or default shall not constitute a waiver of any other right hereunder.

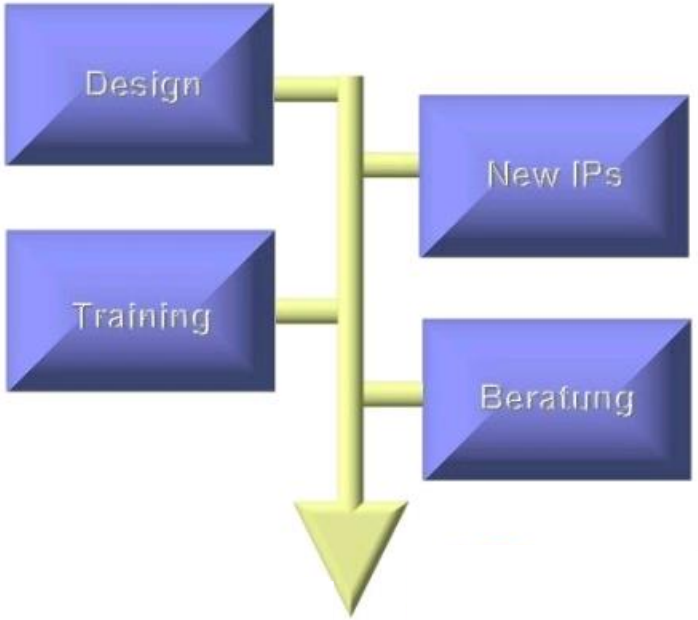
10.6 In the event that any provision of this Agreement is held by a court of competent jurisdiction to be legally ineffective or unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable and the validity of the remaining provisions shall not be affected.

10.7 The article headings throughout this Agreement are for reference purposes only and the words contained therein shall not be construed as a substantial part of this Agreement and shall in no way be held to explain, modify, amplify, or aid in the interpretation, construction or meaning of the provisions of this Agreement.

10.8 BY USING THE IFI IP-MODULE, YOU AND IFI ACKNOWLEDGE THAT YOU AND IFI HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU AND IFI FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN YOU AND IFI, WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN YOU AND IFI RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT, UNLESS YOU HAVE A SEPARATE LICENSE SIGNED BY AN AUTHORIZED IFI REPRESENTATIVE.

Dedicated to FPGA since 1985

- ✓ Quartus® Training Classes
  - QUARTUS-advanced
  - QUARTUS-Expert
- ✓ VHDL Training Classes
- ✓ QSYS Training Classes
  - NIOS®II / QSYS
- ✓ Location
  - Inhouse or Wertheim



- ✓ DesignServices
- ✓ IP
  - CAN Controller
  - Gigabit Ethernet MAC
  - …
- ✓ Consulting

Intel® FPGA Technical Training

INGENIEURBÜRO FÜR IC-TECHNOLOGIE
Franz Sprenger
Kleiner Weg 3
97877 Wertheim
Germany
Tel.: (+49) 9342 / 9608-0
eMail: ifi@ifi-pld.de
http://www.ifi-pld.de

formerly known as: Peter Riekert & Franz Sprenger

