PROCMegaDelay™ IP

User's Guide

September 2008

***GiDEL* products and their generated products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications**

USA
1600 Wyatt Drive Suite 1
Santa Clara,
CA 95054, USA
Tel: 1 - 408 - 969 - 0389
Fax: 1 - 866 - 615 - 6810

sales_usa@GiDEL.com

Worldwide
2 Ha'ilan Street, P.O. Box 281
Or Akiva,
Israel 30600
Tel: +972 - 4610 - 2500
Fax: +972 - 4610 - 2501

sales_eu@GiDEL.com

Web: www.GiDEL.com
info@GiDEL.com

Contents

Figures

Tables

1. Introduction

PROCMegaDelay™ IP is GiDEL's Intellectual Property that provides a simple and convenient way to create large delay lines / frame delays. ***PROCMegaDelay*** eliminates the need to use standard delay lines, which utilize internal FPGA memories. Instead, ***PROCMegaDelay*** uses the on-board memory or SODIMM memory module, thus enabling generation of very large delay lines.

PROCMegaDelay is typically used for 2D / 3D video processing, where very large quantities of data must be stored in memory and extracted later. ***PROCMegaDelay*** makes it possible to compare between two (not necessary subsequent) video frames, or to write video stream as it arrives and read it by frames for further processing.

PROCMegaDelay is simple to use. Its straightforward interface is based on the familiar ***GiDEL PROCMultiPort™*** interface. Once the ***PROCMegaDelay*** has been set up, ***GiDEL PROCWizard™*** automatically connects the required signals to the designer's module. The signals entering the user's module are ready for immediate use.

2. Key Features

PROCMegaDelay was designed to provide an effective and simple means of creating delay lines. ***PROCMegaDelay*** defines the on-board memory (or a part of it) as pipeline storage. Data is inserted on the one end of the pipeline and simultaneously withdrawn from the other end. The length of the pipeline defines the delay time.

The main advantage of ***PROCMegaDelay*** is its ability to work with huge memory banks. The delay memory bank may be on-board DDR memory or memory modules connected to SODIMM sockets. Due to the very large dimensions of these memory banks, the data may be delayed by millions of clocks.

PROCMegaDelay uses cyclic memory pointers. This way, a virtual infinite memory space is created, assuming that the memory bandwidth is not exceeded.

PROCMegaDelay IP key features include:

- User-friendly design with standard input and output
- Simple interface
- Optimized for easy control
- Large memory buffers managed using cyclic memory pointers
- Several outputs may be defined for each ***PROCMegaDelay*** module to output the same data delayed by different factors

3. Setting up PROCMegaDelay

The input and all the outputs of ***PROCMegaDelay*** must be connected to the same subdesign. You should have your target subdesign ready before you start setting up the MegaDelay.

The following should be defined:

- IC(s)
- Main clock and memory clock
- User designs and with appropriate clocks

In the current version of ***PROCDeveloper's Kit***, it is not be possible to put a ***PROCMegaDelay*** in the same memory bank as ***PROCMultiPort***, and vice versa.

To learn how to define subdesigns, please refer to **PROCWizard Version 8. 0 User Manual**.

.

To set up PROCMegaDelay using the PROCWizard:

1. Click **Configure card** in the ***PROCWizard*** toolbar (Figure 1 below) to enter the **Configuration Mode**.

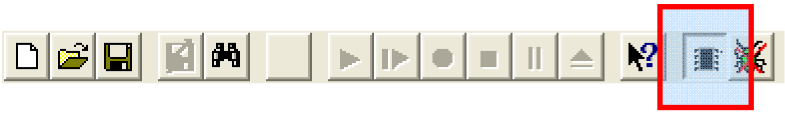


Figure 1 – Configure Card

The **Configuration Mode** is used to build designs in ***PROCWizard***. New items can be added to the design in this mode and existing items can be redesigned.

To learn more about the **Configuration Mode**, please refer to **PROCWizard Version 8. 0 User Manual**.

2. Click the "+" sign to the left of the Designs Card [**proc card type**] item to expand it.

3. Right-click on the name of the desired IC, to access the IC component drop-down menu and select **GiDEL IP Core → MegaDelay**.

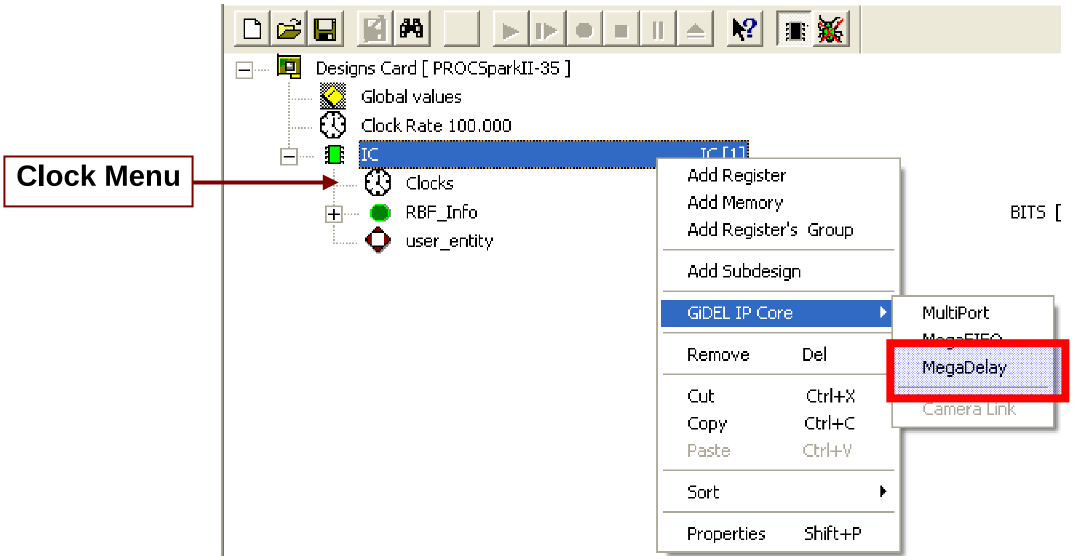
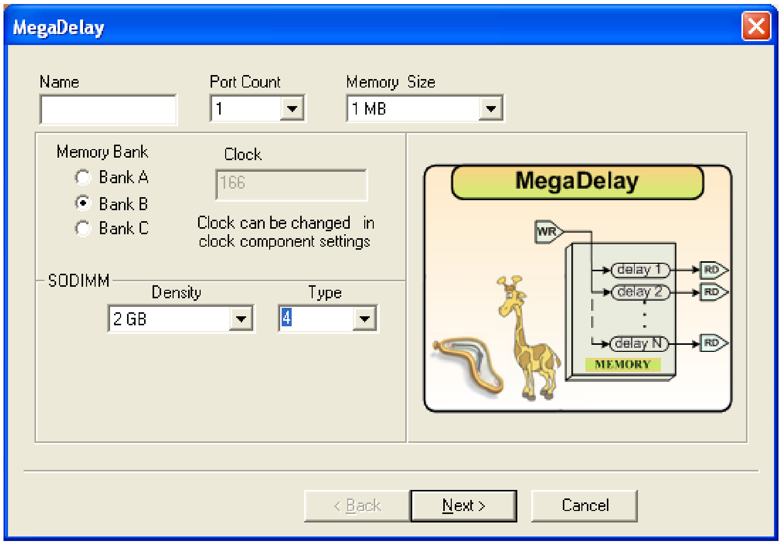


Figure 2 – Accessing MegaDelay in IC Component Drop-Down Menu

The **MegaDelay** dialog box appears.



MegaDelay

Name

Port Count: 1

Memory Size: 1 MB

Memory Bank
○ Bank A
● Bank B
○ Bank C

Clock: 166

Clock can be changed in clock component settings

SODIMM

Density: 2 GB

Type: 4

Figure 3 – MegaDelay Dialog Box

4. In the **MegaDelay** dialog box define the following attributes:

   - The **Name** of the **MegaDelay** memory
   - The output **Port Count** (each output port may output the source data after a different delay)
   - The **Memory Size** to be allocated for the delay

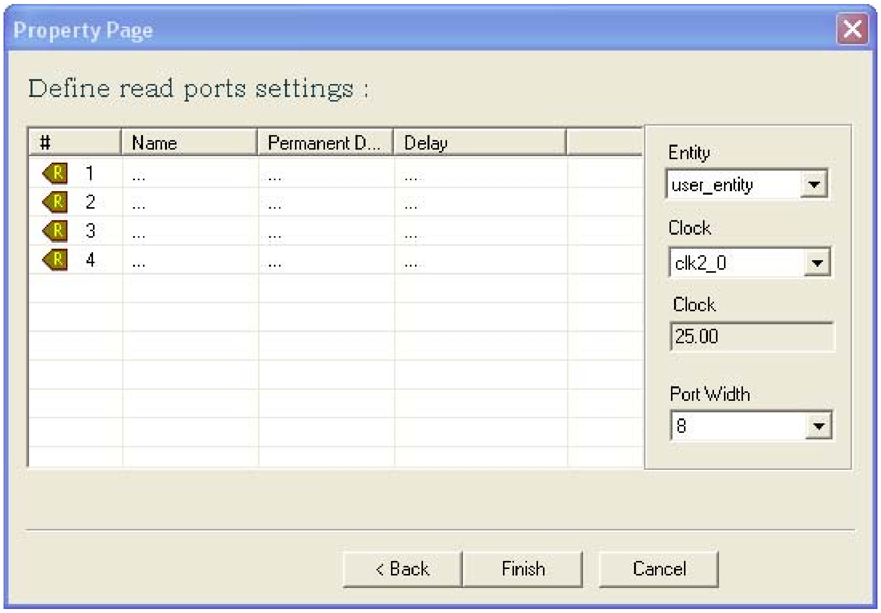
- The **Memory Bank** in which the **MegaDelay** will reside.

  If the selected **Memory Bank** is a SODIMM memory, then a **SODIMM** group box with the fields **Density** and **Type** will appear in the MegaDelay dialog box. (See Figure 3).

  - Click the **Density** arrow to select one of the possible memory densities.
  - Click the SODIMM **Type** arrow to select your SODIMM's memory type. To determine your SODIMM's type, refer to the ***GiDEL SODIMM Type Datasheet*** document. In accordance to the SODIMM's **Type,** the PROCWizard will automatically generate a customized **design**.

The **Memory Clock** rate is in MHz and is set via the clock menu.

5. Click **Next** to continue the configuration of **MegaDelay** in the **Property Page**.



Property Page

Define read ports settings :

#	Name	Permanent D...	Delay
R 1	...	...	...
R 2	...	...	...
R 3	...	...	...
R 4	...	...	...

Entity: user_entity

Clock: clk2_0

Clock: 25.00

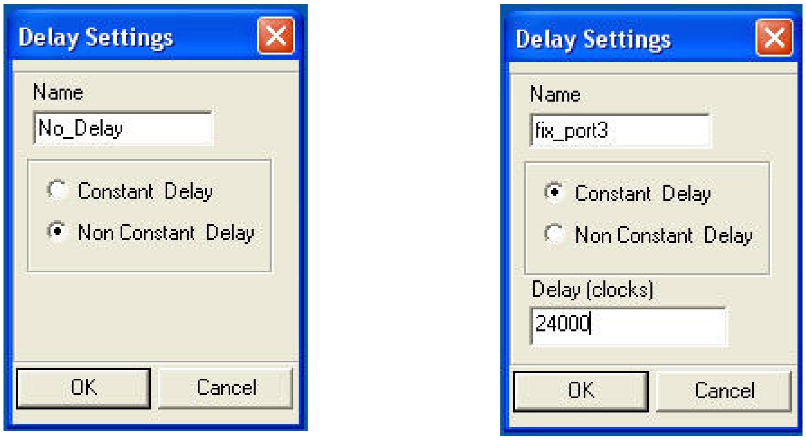
Port Width: 8

Figure 4 – Property Page

6. In the **Property Page** select the required:

   - Target subdesign (**Entity**)
   - Data **Clock** domain
   - Data **Width** in bits

7. Double click in each row to access the **Delay Settings** for every port.



Delay Settings

Name: No_Delay

○ Constant Delay

● Non Constant Delay

Delay Settings

Name: fix_port3

● Constant Delay

○ Non Constant Delay

Delay (clocks): 24000

Figure 5 – Delay Setting Choices

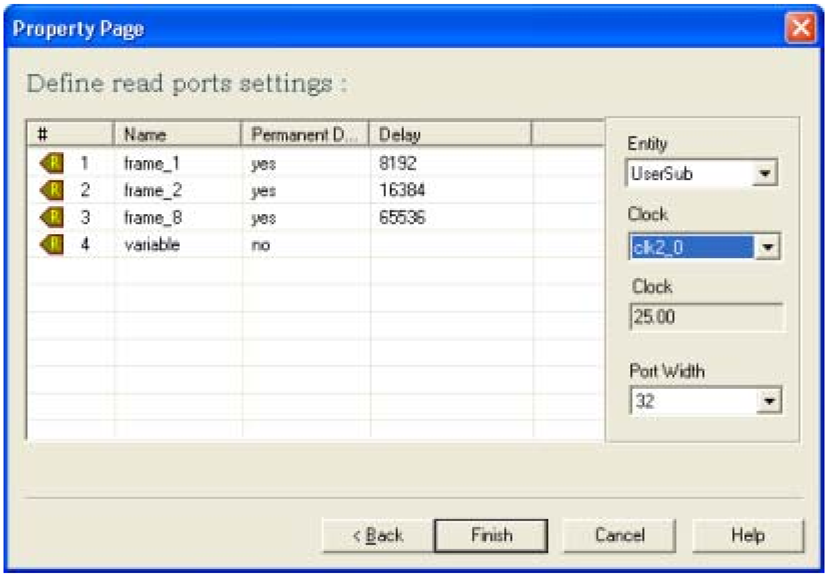
The following settings are available:

- Output port **Name**
- Delay type : **Constant Delay** / **Non Constant Delay**
- **Delay** (in clocks) for **Constant Delay** ports

NOTE	The minimum delay is 8,000 clocks. The maximum delay is determined by the size of the memory in which the **MegaDelay** resides.

If **Non Constant Delay** was chosen for a particular port, the Delay bus of that port will be connected to the target subdesign. This way the user's logic may set a different delay each time the **MegaDelay** is reset.

8. Continue defining delays as required in the **Property Page**



Property Page

Define read ports settings :

#	Name	Permanent D...	Delay
1	frame_1	yes	8192
2	frame_2	yes	16384
3	frame_8	yes	65536
4	variable	no	

Entity: UserSub

Clock: clk2_0

Clock: 25.00

Port Width: 32

Figure 6 – Example of a Filled Property Page

9. Click **Finish** to return to the **Configuration Mode**.

4. PROCMegaDelay in HDL

4.1. General Notes

In the current version of ***PROCWizard***, there is no bandwidth check. The user has to avoid memory throughput override. To calculate the throughput, multiply memory clock rate by memory width and reduce it by 75%.

For example, suppose we have a 32bits wide memory running on 150MHz.

$$\text{Throughput} \approx 0.75*32*150*10^6 = 3.6\text{Gbps}$$

If the memory is of DDR type, the final throughput is multiplied by 2.

$$\text{Throughput} \approx 3.6\text{Gbps} * 2 = 7.2\text{Gbps}$$

The sum of all active ports data rates should be less than the memory throughput.

4.2. Working with PROCMegaDelay

To reset and start PROCMegaDelay:

1. In case of **Non-Constant Delay** port, apply the desired delay on the Delay bus of that port (**delay_MDelay_name_port_name**).
2. Assert the **MDelay_name_reset** pin high for at least 3 periods of the data clock (**MDelay_name_clk**).
3. Wait for **MDelay_name_ready** to go high (usually 10..30 clock periods from the reset fall).

4.3. Writing and Reading data

After **MDelay_name_ready** signal goes high, you may start writing to the source port of **MegaDelay**. Assert **MDelay_name_clk_en** signal high when the data is ready on the source port data bus. Valid data will appear on the read port(s) buses (**data_MDelay_name_port_name**) after the specified number of clocks.

5. PROCMegaDelay Parameters and Signals

The tables below describe the parameters and signals used in the Memory Controller subdesigns. These subdesigns are generated whenever a memory controller is used and are named using the following scheme: **IC_*X*_Bank_*Y*_Ctrl**, where ***X*** is the IC number and ***Y*** is the memory bank name.

These HDL files (.tdf) are automatically generated by ***PROCWizard*** whenever the SDRAM controller is used. The signals that are driven from the user logic are automatically added to the user subdesigns, with the same name.

NOTE	1. All GiDEL IPs should be connected using automatic generation with ***PROCWizard***. 2. The parameters and the signals are defined automatically and there is no need to change them manually. The best way to change them is to define a new design in ***PROCWizard*** and generate a new HDL code. 3. There are a number of the signals and parameters, which are not described here. They are generated automatically to suit current design. The user should not change them in any case. 4. ***PROCMegaDelay*** IP does not interfere in the throughput control. Be careful not to exceed memory throughput and use **MDelay_name_ready** signal.

5.1. Parameters for PROCMegaDelay

Parameter name	Purpose	Comments
ADDR_WIDTH	Memory address bus width (in bits)	For example, if we define a 16MB MegaDelay and the memory cell is 4 bytes wide, then we have 4M address space and 22-bit address bus: ADDR_WIDTH = Log2(16M / 4) = 22
DATA_WIDTH	Port data bus width (in bits)	
MEM_WIDTH	Memory word width (in bits)	
NUMBER_OF_READ_PORTS	The number of read ports	The number of read ports is the number of different delay lines in the design.

Table 1 - PROCMegaDelay Parameter Names

5.2. Signal/Bus Names for PROCMegaDelay

Signal / Bus Name * / **	Direction	Purpose	Comments
clrn	INPUT	Global reset signal	Global reset signal is active low.
mem_clk	INPUT	Memory clock	
MDelay_name_clk	INPUT	MegaDelay clock	All MegaDelay ports are synchronized with this clock.
MDelay_name_reset	INPUT	MegaDelay reset	Reset pulse is active high. Should be at least 3 clocks wide. Resets the MegaDelay.
MDelay_name_ ready	OUTPUT	Ready signal	When high: MegaDelay unit is ready to accept and deploy data
MDelay_name_ clk_en	INPUT	Clock enable	MegaDelay unit will accept and deploy data upon each clock when this signal is high
Delay_MDelay*_ name_port_name**	INPUT	Delay time (in clocks)	For ports with non-fixed delay, this bus sets the desired delay time. Place the desired number of clocks to delay on this bus before resetting the MegaDelay
Data_MDelay_ name_src	INPUT	Source data	Data to delay
Data_MDelay_ name_port_name	OUTPUT	Delayed data	

Table 2 - MegaDelay Signal and Bus Names

* **port_name** is the name given to the MegaDelay port during the definition process in ***PROCWizard***.

** **Mdelay_name** is the name given to the MegaDelay module during the definition process in ***PROCWizard***.

6. Accessories

6.1. GiDEL PROC Developer's Kit™

GiDEL PROC Developer's Kit is a set of building blocks designed for fast, high-productivity system development. It is a complete system solution including boards, software tools, IPs and optional daughterboards. The major software tools and IPs are detailed in the following paragraphs.

6.2. GiDEL PROCWizard™

GiDEL PROCWizard is a convenient developer's environment that can automatically generate the Hardware/Software interface for your application. This interface includes:

- **Application driver (C class)**
  The driver can be generated for Windows or Linux environment. The driver is built in two layers:
  - The lower layer (the **Proc** class) implements basic board functionality, such as FPGA loading, DMA interfaces, Interrupt Service Routines, board clocking system setups, board information acquisition (FPGA size, speed grade etc). The **Proc** class is supplied with the ***PROCWizard***.
  - The higher layer is automatically generated by the ***PROCWizard***. This class inherits from the **Proc** class and implements all the application-specific functionality. It loads the Stratix II device, sets up the board clocking and initializes all the class members to allow simple access to the board application from the user space.

- **HDL code: (Verilog, VHDL or AHDL)**
  PROCWizard can automatically generate the following features:
  - Interface module/entity that communicates with the software driver.
  - PROCMultiPort (on-board memory controller) instantiations
  - Basic PLLs to control external memories
  - Top-level design connecting all the above instantiations, with user's modules/entities and the on-board local bus and memories
  - Device constraints including FPGA pin-out, pins' power voltage (VCCIO) Quartus operation recommendations etc.

In addition, the ***PROCWizard*** can generate documentation in **.html** or **.doc** format describing in detail the generated features.

PROCWizard also enables you to test and debug the design in a PC environment. Using ***PROCWizard*** you can access PROC boards with a structural browser and macros/scripts. You may load / save and compare memory files to check data transfers and access the registers / memories defined in the design in real time.

For more information, please refer to the **PROCWizard User Manual 8.0**.

6.3. GiDEL PROCMultiPort™

PROCMultiPort completes the PROC board features and provides an advanced controller for the on-board memories. This controller has up to 16 ports; each port features a simple access (FIFO-like or random). All ports are connected to the same memory domain and can be accessed independently and simultaneously with individual clock domains and data widths. ***PROCMultiPort*** segmented mode provides the ability to logically enlarge the FPGA memory size.

The innovative ***PROCMultiPort*** concept enables new design methodologies. It can replace many large and complicated designs, thus reducing development effort. For example, it can replace swappable double buffers or implement multiple logical memories in the same physical memory.

For more information, please refer to the ***PROCMultiPort*** **IP User Guide**.

6.4. GiDEL PROCMegaFIFO™

PROCMegaFIFO is GiDEL's IP that provides a simple and convenient way to transfer data to / from **GiDEL *PROC*** boards. With ***PROCMegaFIFO***, data may be transferred between the host PC and user's subdesigns, or between subdesigns, using the on-board memory as a very large FIFO.

PROCMegaFIFO eliminates the need to take care of synchronization when transferring data between designs. The software no longer needs to respond to the hardware in real-time, and hardware designs may now transfer data in bursts and withdraw it in a continuous stream.

PROCMegaFIFO uses special arbitration techniques when transferring data between the host PC and user's subdesigns. These techniques prevent memory overflows / underruns, thus using the maximum available bandwidth for data transfers.

Request and Acknowledge signals ensure correct data transfers. On the software side, the Proc class methods perform automatic initialization of the FIFO logic and enable easy data transfers using DMA.

For more information, please refer to the **PROCMegaFIFO IP User Guide**.

.

7. References

7.1. REFERENCES:

- **PROCWizard Version 8. 0 User Manual**
- **PROCMultiPort IP User Guide**
- **PROCMegaFIFO IP User Guide**

8. Appendix

8.1. User Guide History

Date	Changes
Feb 2006	Rev.01 User Guide
Sept. 2008	Addition of SODIMM **Density** and **Type**