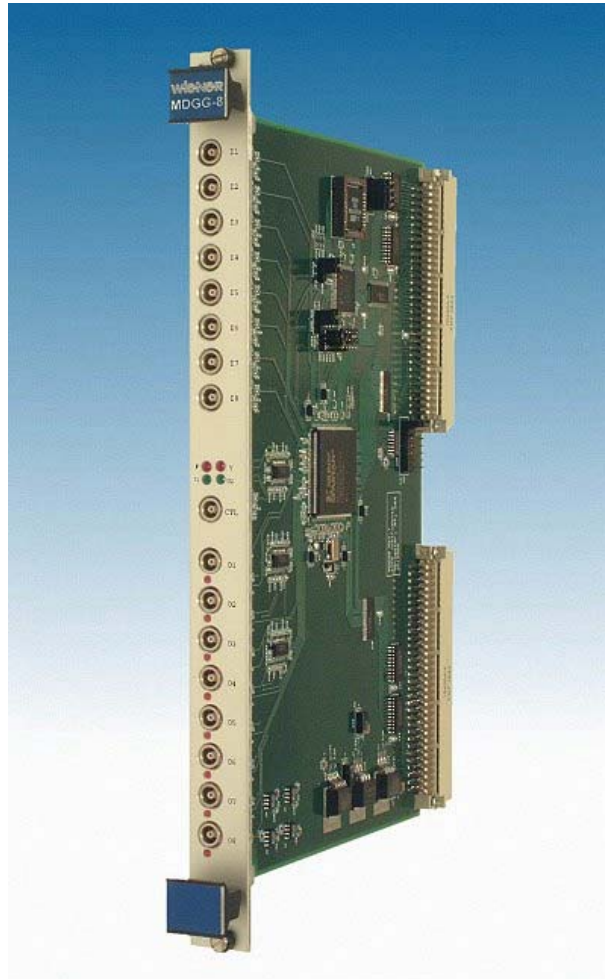


# MDGG-8



## User Manual

## General Remarks

The only purpose of this manual is a description of the product. It must not be interpreted a declaration of conformity for this product including the product and software.

**W-Ie-Ne-R** revises this product and manual without notice. Differences of the description in manual and product are possible.

**W-Ie-Ne-R** excludes completely any liability for loss of profits, loss of business, loss of use or data, interrupt of business, or for indirect, special incidental, or consequential damages of any kind, even if **W-Ie-Ne-R** has been advises of the possibility of such damages arising from any defect or error in this manual or product.

Any use of the product which may influence health of human beings requires the express written permission of **W-Ie-Ne-R**.

Products mentioned in this manual are mentioned for identification purposes only.

Product names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies.

No part of this product, including the product and the software may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means with the express written permission of **W-Ie-Ne-R**.

MDGG-8 is designed by JTEC Instruments.

## Table of contents:

1. General Description .....	4
Hardware features .....	4
Release firmware features (preliminary) .....	4
2. User interface .....	6
3. Customization .....	6
4. REGISTERS .....	6
4.1 Delay and Gate Length Registers, DelayRegn and GateRegn (0x40 – 0x7C) .....	6
4.2 Global Register, GlobalReg (0x4) .....	6
4.3 Auxiliary Mask Register, AuxReg (0x8) .....	7
4.4 VME write-only action toggle register, ActionReg (0x80) .....	7
4.5 FGG Configuration Register, FGGConfig (0x84) .....	8
4.6 Scaler Configuration Register, SclrConfig (0x88) .....	8
4.7 FGG Trigger Selector Registers (0x8C and 0x90) .....	9
4.8 NIM Output Source Selector Registers (0x94 and 0x98) .....	10
4.9 Gate Reset Signal Selector Registers (0x9C and 0xA0) .....	10
4.10 Scaler Input Selector Register (0xA4 and 0xA8) .....	11
4.12 Combinatorial Gates Mask Registers (0xAC and 0xB0) .....	12
4.13 caler Data Registers (0x100 – 0x13C) .....	12
4.14 Coincidence Register Data Word (0x140) .....	13
5. Summary of the VME Address Space of MDGG8 .....	14

## 1.

## GENERAL DESCRIPTION

### Hardware features

1. 8+1 NIM inputs
2. 8 NIM outputs
3. 12 diagnostic LED's, driven by signal stretchers
4. XC3S400 FPGA with 36kB of block RAM
5. Two FPGA configuration files stored in a flash memory – the FPGA can boot from either one (selection by a jumper).
6. The flash memory can be reprogrammed via VME.
7. A24 addressing mode
8. D32 data width
9. IRQ capable, with the IRQ level selected by a jumper.
10. Start-up base address set by 5 jumpers.

### Release firmware features (preliminary)

1. Eight 32-bit flexible digital (8ns granularity) gate generators (FGG), configurable individually as
  - a. digital delay and gate generators (DGG), configurable individually as
    - i. non-retriggerable delay and gate generators (DGG)s,
    - ii. delay + retriggerable gate generators (RDGG), or
    - iii. pulse generators (PG)
  - b. set-reset gates (SRG), or
  - c. prescaler gates, configurable individually as
    - i. 1/n prescalers (PSG) or
    - ii. 1-1/n complementary prescalers (CPSG)
2. Eight scaler devices, configurable individually as
  - a. regular gated 32-bit scalers (GSC) or
  - b. latchable 32-bit scalers (LSC), each with an individual 1kx32 FIFO storage
3. One 8-bit coincidence register (CREG).
4. Four combinatorial gates (CG2x8) of two-fold ORs of eightfold ANDs
5. Up-to 20-fold individual input multiplexers for flexible gates and scalers, including
  - a. eight NIM inputs

- b. eight end-of-output signal (trailing edge) of the eight flexible devices, and
  - c. four outputs of the four combinatorial gates
6. 17-fold multiplexer for the scaler gating/latching signal, including
  - a. nine NIM inputs and
  - b. eight output signals of the eight flexible gates
7. 9-fold multiplexer for the scaler reset, including all nine NIM inputs.
8. Veto of trigger signals of individual flexible gates with a 9-fold selector of the common vetoing signal – any of the eight regular or the one control NIM input.
9. VME triggering and resetting of flexible gates.
10. VME reset of individual scalers
11. Simple or block (BLT32) readout of the firmware ID, the content of the configuration registers, the eight scalers, one coincidence register, the eight latch multiplicities of the individual latched scalers (number of 32-bit words stored in individual FIFOs) and the contents of the eight latched scaler FIFOs.
12. 20-fold individual output selectors associated with every NIM output, including
  - a. eight output signals of the eight FGGs
  - b. trailing edges of the eight output signals of the FGGs, and
  - c. four output signals of the four combinatorial gates
13. Individually selectable output polarity (invert) for the eight NIM outputs
14. Twelve diagnostic LEDs (driven by stretchers) indicating
  - a. the 3-state status of the FPGA (top left red 3mm LED) :
    - i. on – FPGA not configured
    - ii. flashing – flash memory being programmed
    - iii. off – MDD8 configured
  - b. the VME access of MDGG8 (top right red 3mm LED)
  - c. the receipt of a signal at the control NIM input (bottom left 3mm green LED)
  - d. the receipt of a signal at any of the eight regular NIM inputs (bottom right 3mm green LED)
  - e. the presence of a signal at an individual NIM output (eight 2mm LEDs).

## 2. USER INTERFACE

The desired configuration of MDGG8 is achieved by storing the configuration data in a set of approximately forty registers via VME D32 A24 “write” commands.

## 3. CUSTOMIZATION

MDGG8 can be readily customized within the constraints set by the hardware resources. In fact, it may be considered a 9-input/8-output universal logical module for which the user can develop his own firmware using the free XILINX WebPack software.

## 4. REGISTERS

### 4.1 Delay and Gate Length Registers, DelayRegn and GateRegn (0x40 – 0x7C)

There are 16 32-bit registers for storing the desired values for delay and gate lengths of the eight FGGs, FGG1 – FGG8. The VME address offsets are given by the equations

$$\text{Offset(DelayRegn)} = 56 + 8*n, \text{ where } n=1-8 \quad (0x38 + 8*n \rightarrow 0x40, 0x48, 0x50 \dots\dots)$$

$$\text{Offset(GateRegn)} = 60 + 8*n, \text{ where } n=1-8. \quad (0x3C + 8*n \rightarrow 0x44, 0x4C, 0x54 \dots\dots)$$

Note that the GateReg registers are used also for storing prescale factors for prescale gates.

### 4.2 Global Register, GlobalReg (0x4)

The (32-bit) global register stores information on four global signals - common to classes of devices. These four global signals include a common veto gate for all FGGs that are configured to be veto-sensitive, a common gate signal for all gated scalers, a common latch signal for all latched scalers, and a master reset signal for FGGs, scalers, and the coincidence register. The identification of sources of global signals is achieved via 5-bit words addressing the respective signal selectors (one of twenty input signals).

$$\text{Offset(GlobalReg)} = 4$$

The structure of the global register is shown in the table below

Bits 24 – 28	Bits 16 - 20	Bits 8 – 12	Bits 0 – 4
SelfMReset	SelfScLrLatch	SelfScLrGa	SelfFGGVeto

Values 0 – 21 of any of the selector words represent

0	none selected
1 – 8	I1-I8, NIM inputs
9 – 16	FG1-FG8, gates generated by the 8 FGGs
17-20	CG1-CG4, combinatorial gates
21	CTL, NIM CTL input

### 4.3 Auxiliary Mask Register, AuxReg (0x8)

The 14-bit mask register stores information on the reset mask for FGGs and the coincidence register, as well as on the gate source for the latter. The mask bits determine, whether the individual FGGs or the CREG are reset by the Master Reset signal

Offset(AuxReg) = 8

The structure of the Auxiliary Mask Register is shown in the table below

Bit 13	Bits 8 - 12	Bits 0 – 7
CRegMResetMask	SelCRegGa	FGGMResetMask

Values 0 – 20 of any of the CReg gate selector word SelCRegGa represent

0 – 7	I1-I8, NIM inputs
8 – 15	FG1-FG8, gates generated by the 8 FGGs
16-19	CG1-CG4, combinatorial gates
20	CTL, NIM CTL input

Note that the coincidence register must be cleared, before it is ready to accept the next gate.

### 4.4 VME write-only action toggle register, ActionReg (0x80)

The MDGG8 firmware allows one to trigger any selection of FGGs and to reset any desired selection of FGGs, scalars, or the CREG by writing 1s to the respective bits of a toggle register (bits clear automatically in 16 ns).

Offset(ActionReg) = 128 (0x80)

The structure of the action register is shown in the table below

<b>Bit 24</b>	<b>Bits 16 - 23</b>	<b>Bits 8 - 15</b>	<b>Bits 0 - 7</b>
<b>ResCREG</b>	<b>TrigFGG1-8</b>	<b>ResSc1r1-8</b>	<b>ResFGG1-8</b>

Not that the actual reset signals for individual gates, scalers, and the coincidence register are three-fold logical ORs of individual VME resets, and individual and global resets derived from external sources.

#### 4.5 FGG Configuration Register, FGGConfig (0x84)

Any FGG can be configured to be either a non-retriggerable DGG, a set-reset gate, SRG, a pulser, PG, a retriggerable DGG, RDGG, a 1/n prescaler, PSG, or a complementary 1-1/n prescaler, CPSG. Furthermore, any FGG can be configured to be subject to a common veto signal. The actual configuration of a single FGG is identified by a 4-bit word, with bit 3 (value 8) serving as a veto mask. Setting bit 3 = 1 sensitizes an FGG to the common veto signal.

Offset(FGGConfig) = 132 (0x84)

The structure of the FGGConfig register is shown in the table below

<b>28-31</b>	<b>24-27</b>	<b>20-3</b>	<b>16-19</b>	<b>12-15</b>	<b>8-11</b>	<b>4-7</b>	<b>Bits 0-3</b>
<b>FGG8</b>	<b>FGG7</b>	<b>FGG6</b>	<b>FGG5</b>	<b>FGG4</b>	<b>FGG3</b>	<b>FGG2</b>	<b>FGG1</b>

The values of the 3 least significant bits (bits0-2) of the individual configuration words represent

- 0 FGG off
- 1 DGG
- 2 SRG
- 3 PG
- 4 RDGG
- 5 PSG
- 6 CPSG

- Bit3=1 FGG is subject to the common veto gate
- Bit3=0 FGG is not subject to the common veto gate.

#### 4.6 Scaler Configuration Register, Sc1rConfig (0x88)

Any scaler can be configured to function either as a gated scaler, a latchable scaler, or a gated latchable scaler. In a gated scaler mode, the scaler counts trigger signals when the

gate is active and suspends counting when the gate is off. In a latchable mode, scaler is active as long as the 1kx32 storage FIFO is not full and increments by 1 with every trigger signal. The state of the scaler is stored in FIFO upon the receipt of a latching signal. At the same time a latch multiplicity counter is incremented so that its content represents the number of words written into the FIFO. Both FIFO and the multiplicity counter can be read out. In the gated latchable mode, the counter is incremented only when the gate is active.

Offset(SclrConfig) = 136 (0x88)

The structure of the SCLRConfig register is shown in the table below

<b>28-30</b>	<b>24-26</b>	<b>20-22</b>	<b>16-18</b>	<b>12-14</b>	<b>8-10</b>	<b>4-6</b>	<b>Bits 0-2</b>
<b>SCLR8</b>	<b>SCLR7</b>	<b>SCLR6</b>	<b>SCLR5</b>	<b>SCLR4</b>	<b>SCLR3</b>	<b>SCLR2</b>	<b>SCLR1</b>

Significance of the two least significant configuration bits for an individual scaler is as follows:

“00”	scaler off
“01”	gated mode
“10”	latched mode
“11”	gated latched mode with count suspended when gating signal is off

Bit 3 of the configuration word determines whether an individual scaler responds to the Master Reset signal (1 – clears upon MReset, 0 – no action on MReset).

#### 4.7 FGG Trigger Selector Registers (0x8C and 0x90)

The trigger source for an individual FGG is identified by a 5-bit word. These words for all 8 FGGs are stored in two registers as follows:

At Offset = 0x8C

<b>Bits24-28</b>	<b>Bits 16 – 20</b>	<b>Bits 8 – 12</b>	<b>Bits 0 - 4</b>
<b>TrigSelFGG4</b>	<b>TrigSelFGG3</b>	<b>TrigSelFGG2</b>	<b>TrigSelFGG1</b>

At Offset = 0x90

<b>Bits24-28</b>	<b>Bits 16 – 20</b>	<b>Bits 8 – 12</b>	<b>Bits 0 - 4</b>
<b>TrigSelFGG8</b>	<b>TrigSelFGG7</b>	<b>TrigSelFGG6</b>	<b>TrigSelFGG5</b>

Values 0 – 20 of any of the FGG trigger selector word TrigSelFGGn represent

0 – 7	I1-I8, NIM inputs
8 – 15	TrEdge1-TrEdge8, trailing edges of gates generated by the 8 FGGs
16-19	CG1-CG4, combinatorial gates
20	CTL, NIM CTL input

#### 4.8 NIM Output Source Selector Registers (0x94 and 0x98)

Every active Flexible Gate Generator generates two signals, the gate and its trailing edge marker (TrEdge). Further, every combinatorial gate generates one signal representing the result of the logical OR of 8-fold logical ANDs of selected input signals. Each of the above signals can be routed to any of the NIM outputs (O1-O8) of MDGG8, by writing a proper 5-bit code into one of two NIM Output Source Selector Registers. The structure of these registers is as follows:

At Offset = 0x94

<b>Bits24-28</b>	<b>Bits 16 – 20</b>	<b>Bits 8 – 12</b>	<b>Bits 0 - 4</b>
<b>NIMOutSel 4</b>	<b>NIMOutSel 3</b>	<b>NIMOutSel 2</b>	<b>NIMOutSel1</b>

At Offset = 0x98

<b>Bits24-28</b>	<b>Bits 16 – 20</b>	<b>Bits 8 – 12</b>	<b>Bits 0 - 4</b>
<b>NIMOutSel 8</b>	<b>NIMOutSel 7</b>	<b>NIMOutSel 6</b>	<b>NIMOutSel 5</b>

Values 0 – 19 of any of the output source selector word NIMOutSel n represent

0 – 7	Gate1-Gate8, gate signals of the 8 FGGs
8 – 15	TrEdge1-TrEdge8, trailing edges of gates generated by the 8 FGGs
16-19	CG1-CG4, combinatorial gates

#### 4.9 Gate Reset Signal Selector Registers (0x9C and 0xA0)

Any FGG configured as a Set-Reset latch is triggered (set) by a trigger signal as defined by its 5-bit code in the Trigger Source Selector Register. It is reset either by a global reset signal (subject to FGG Reset Mask), by its individual VME reset signal, or by a signal identified by a 5-bit code in one of the two Gate Reset Signal Selector Registers. The latter registers have the following structure:

At Offset = 0x9C

Bits24-28	Bits 16 – 20	Bits 8 – 12	Bits 0 - 4
StopSelFGG4	StopSelFGG3	StopSelFGG2	StopSelFGG1

At Offset = 0xA0

Bits24-28	Bits 16 – 20	Bits 8 – 12	Bits 0 - 4
StopSelFGG8	StopSelFGG7	StopSelFGG6	StopSelFGG 5

Values 0 – 20 of any of the FGG tstop selector words **StopSelFGGn** represent

0 – 7	I1-I8, NIM inputs
8 – 15	TrEdge1-TrEdge8, trailing edges of gates generated by the 8 FGGs
16-19	CG1-CG4, combinatorial gates
20	CTL, NIM CTL input

#### 4.10 Scaler Input Selector Register (0xA4 and 0xA8)

Input source for an individual scaler is identified by a 5-bit word. These words for all 8 scalers are stored in two registers as follows:

At Offset = 0xA4

Bits24-28	Bits 16 - 20	Bits 8 – 12	Bits 0 - 4
SclrInSel 4	SclrInSel 3	SclrInSel 2	SclrInSel 1

At Offset = 0xA8

Bits24-28	Bits 16 - 20	Bits 8 – 12	Bits 0 - 4
SclrInSel 8	SclrInSel 7	SclrInSel 6	SclrInSel 5

Values 0 – 20 of any of the scaler input selector word **SclrInSeln** represent

0 – 7	I1-I8, NIM inputs
8 – 15	TrEdge1-TrEdge8, trailing edges of gates generated by the 8 FGGs
16-19	CG1-CG4, combinatorial gates
20	CTL, NIM CTL input

#### 4.12 Combinatorial Gates Mask Registers (0xAC and 0xB0)

Each of the four combinatorial gates of MDGG8 represent a two-fold logical OR of (up-to 8-fold) logical ANDs of selected NIMn (n=1-8) inputs. The selection of active NIM inputs is achieved via 8-bit AND masks (AMASK(1:4,1:2)), such that the active inputs have their respective bits set to 1, while inactive inputs have bits reset to 0. Thus, the logical equation for the n-th combinatorial gate CGn reads:

$$CG_n = \{ [AMASK(n,1) \text{ AND } NIM] = AMASK(n,1) \} \\ \text{OR } \{ [AMASK(n,2) \text{ and } NIM] = AMASK(n,2) \},$$

where NIM is an 8-bit word representing the status of the 8 NIM inputs.

Note that any inactive term in the above equation must have all mask bits set to 1 and not to 0. Obviously, in the latter case, the result would be true for no NIM inputs present.

The 8 AMASK words are stored in two registers in the following manner

At Offset = 0xAC

Bits24-31	Bits 16 - 23	Bits 8 – 15	Bits 0 – 7
AMASK(2,2)	AMASK(2,1)	AMASK(1,2)	AMASK(1,1)

At Offset = 0xB0

Bits24-31	Bits 16 - 23	Bits 8 – 15	Bits 0 – 7
AMASK(4,2)	AMASK(4,1)	AMASK(3,2)	AMASK(3,1)

#### 4.13 Scaler Data Registers (0x100 – 0x13C)

MDGG8 stores scaler data in respective 1kx32 FIFOs, for read-back via VME commands. The number of words in FIFOs is stored in respective hit multiplicity registers. For scalers operated in gated mode, only one word is stored in the data FIFO with the hit multiplicity being always 1. A gated scaler must be reset, before it is ready to accept subsequent gate. For latched scalers, FIFOs contain the latched states of the counters. A latched scaler is active as long as its associated FIFO is not full, subject to the gating signal, if applicable (gated latched mode). VME address offsets for the Scaler FIFOs and Hit Multiplicities are as follows:

$$\text{Offset(FIFO}_n) = 0xFC + n * 4, n=1-8$$

Offset(HitMultn) =  $0x11C + n*4$ ,  $n=1-8$

#### 4.14 Coincidence Register Data Word (0x140)

The state of the coincidence register CReg is available for readout at VME offset 0x140. Note that the register must be cleared, before it is ready to accept a new gate.

## 5. SUMMARY OF THE VME ADDRESS SPACE OF MDGG8

Offset	Register	Access Type
0x000	Firmware ID	Read only
0x004	Global Register	Read/write
0x008	Auxiliary Register	Read/write
0x040	Delay FGG1	Read/write
0x044	Gate FGG1	Read/write
0x048	Delay FGG2	Read/write
0x04C	Gate FGG2	Read/write
0x050	Delay FGG3	Read/write
0x054	Gate FGG3	Read/write
0x058	Delay FGG4	Read/write
0x05C	Gate FGG4	Read/write
0x060	Delay FGG5	Read/write
0x064	Gate FGG5	Read/write
0x068	Delay FGG6	Read/write
0x06C	Gate FGG6	Read/write
0x070	Delay FGG7	Read/write
0x074	Gate FGG7	Read/write
0x078	Delay FGG8	Read/write
0x07C	Gate FGG8	Read/write
0x080	Action Register	Write only
0x084	FGG Configuration	Read/write
0x088	Scaler Configuration	Read/write
0x08C	FGG Input Selector A	Read/write
0x090	FGG Input Selector B	Read/write
0x094	NIM Output Selector A	Read/write
0x098	NIM Output Selector B	Read/write
0x09C	FGG Stop Selector A	Read/write
0x0A0	FGG Stop Selector B	Read/write
0x0A4	Scaler Input Selector A	Read/write
0x0A8	Scaler Input Selector B	Read/write
0x0AC	Logical AND Mask A	Read/write
0x0B0	Logical AND Mask B	Read/write
0x100	Scaler Data 1	Read only
0x104	Scaler Data 2	Read only
0x108	Scaler Data 3	Read only
0x10C	Scaler Data 4	Read only
0x110	Scaler Data 5	Read only
0x114	Scaler Data 6	Read only
0x118	Scaler Data 7	Read only
0x11C	Scaler Data 8	Read only



0x120	Scaler Latch Multiplicity 1	Read only
0x120	Scaler Latch Multiplicity 2	Read only
0x120	Scaler Latch Multiplicity 3	Read only
0x120	Scaler Latch Multiplicity 4	Read only
0x120	Scaler Latch Multiplicity 5	Read only
0x120	Scaler Latch Multiplicity 6	Read only
0x120	Scaler Latch Multiplicity 7	Read only
0x120	Scaler Latch Multiplicity 8	Read only
0x140	Coincidence Register	Read only