RIFTEK
SENSORS & INSTRUMENTS

We implement ideas

D-MOT-10 FPGA IP CORE FOR MOTION DETECTION IN VIDEO STREAM IMAGES

from RIFTEK Ltd.
Cores
IMAGING

10.04.2013

(ENG)

TABLE OF CONTENTS

INTRODUCTION

The D-MOT-10 FPGA IP core for motion detection in video stream images is a completed module intended to be used in vision systems for various applications (security systems, specialized systems, etc.) that are based on FPGA. The core represents a simple data exchange interface that ensures easy integration into various systems. The algorithms are perfectly suitable for motion detection of ground, surface and aerial objects of any type. The implemented algorithms provide detecting of small-size and low-contrast objects against a complex background. When an object is detected the core assigns it a unique number and determinates the movement direction. This makes it easy to track the object behavior and build efficient analytical system (perimeter defense, road traffic control and etc.) When a moving object is detected its motion path is calculated. In case the object disappears from the field of vision (for instance, behind an obstacle) the core predicts the path and detects the object when it re-appears. It is possible to detect up to 128 moving objects simultaneously. Thus, the D-MOT-10 IP core is a completed module that allows its use in any FPGA projects for vision systems.

CORE VERSIONS

Versions of the D-MOT-10 IP core are shown in Table 1.

Table 1 – Versions of the D-MOT-10 IP core.

Version	Notes
1.0	First version of the IP core. Implements a modified high-performance algorithm of moving objects detection in the video sequence images.

BASIC CHARACTERISTICS

The D-MOT-10 IP core ensures detection of moving objects in video sequence images. The basic characteristics of the core are given in Table 2.

Table 2 – Basic characteristics of the D-MOT-10 IP core.

Parameter	Values and notes
Detecting mode	Automatic detection of moving objects by comparing the sequence of frames in the video stream.
Maximum number of objects detecting simultaneously	The core provides simultaneous detection of up to 128 moving objects in video sequence. Range of unique numbers assigned to the objects is from 0 to 127 incl.
Maximum size of detecting object	The maximum size of detecting object is limited by the size of processed area (the size of the frame buffer).
Minimum size of detecting object	The core ensures stable detecting of moving objects of up to 8x8 pixels size in an image area.
Maximum and minimum speed of detecting object	For a sudden change of the position of tracking object, the speed is no more than 30 pixels per frame in any direction and no less than 0.1 pixels per frame in any direction.
Minimum contrast of detecting objects	The core ensures stable detection of moving objects with contrast from 10%.
Maximum size of frames	The core provides the work with frame sizes up to 720x576 pixels. At customer's request, the core can be synthesized for frame sizes up to 5120x5120 pixels. When it is needed to process big images it is recommended to use several cores simultaneously.

Parameter	Values and notes
Elongation of detecting objects	Tracking objects can have any configuration.
Output coordinates accuracy	Output coordinates of tracking objects are obtained with 1 pixel accuracy.
Time lag of output coordinates	At the frame refresh rate of 25 Hz the core calculates the extrapolated coordinates of the object before the end of the next frame in the video sequence (delay is 1 frame).
Time from the core initialization to stable detecting mode	After the core initialization it needs 2-3 minutes to adapt to the current conditions of observation, main condition - immobility of the camera view.
Format of processed images	The core works with images having color palette of 256 colors (grayscale) – 8 bits/pixel.
Functioning conditions	The observed scene contained in the image sequence should be fixed (the camera view is stable and does not move in time).

Note: the above data are given for FPGA Xilinx Spartan-6XC6SLX150T-3.

USEFUL LINKS

D-MOT-10 IP Core documentation can be downloaded by the following links:
D-MOT-10 Datasheet, Programmer Manual

IP Cores Catalog

The order can be made by e-mail: info@riftek.com, sales@riftek.com.

PART 1 OVERVIEW OF THE CORE

DESCRIPTION AND OPERATION PRINCIPLE

The core is a complete module intended for use in FPGA system vision projects. It works as follows. The core is connected to the interface controller of DDR memory via a standard interface (controller provided by Xilinx Company) from which it reads the current images of video sequence and where it writes the results. The designer should ensure refreshment of image frames in the DDR memory according to multi-buffer principle (multiple buffering). Before operation is started, the core should be configured (configuration register values need to be written). Observed scene (the scene in the images of the video sequence) should be fixed. After initialization the core (core switching) it needs some time (2-3 minutes) to adapt to the observed scene. During this time, the core algorithms initialize filters and form the map of constantly moving objects (e.g. treetops wave in the wind). During this time false detections are possible. After all core filters are adapted all moving objects are detected with the high degree of reliability. Herein only objects having a specific trajectory (not accidental movement in an image frame) are detected. When a moving object is detected the core algorithms estimate its size and assign to it a unique numerical code that can be used in the further processing outside the core. The core gives the information about moving objects for each frame in the video sequence. For each detected moving object the core gives the coordinates of the corners of the detecting strobe in the image plane, numerical number assigned to the object, as well as its movement direction, all of that allows to build effective perimeter control and traffic analysis systems on the basis of the core. The core provides detection of up to 128 simultaneously moving objects. The output coordinate is determined with a delay equal to 1 frame.

APPLICATION FIELD AND PLATFORMS USED

The core can be used in any vision applications such as security video surveillance systems and in special robotic vision systems. A simple interface of data exchange with the core makes it possible to integrate it easily into any FPGA project. The core is made available to customer with specification of FPGA type. The core is delivered to the customer in a synthesized form for a specific type of FPGA under license. The core was developed for FPGA's of Xilinx Company and supports the following of their families:

Artix-7, Artix-7Q;
Kintex-7, Kintex-7Q;
Virtex-7, Virtex-7Q;
EasyPath-7;
Virtex-6, Virtex-6Q;
EasyPath-6;
Spartan-6;
EasyPath;
Virtex-5Q, Virtex-5QV;
Virtex-4Q, Virtex-4QV;
XA Spartan-6;
XA Spartan-3A, XA Spartan-3A DSP, XA Spartan-3E.

PROCEDURE OF USE

The use of the core requires that a FPGA project be equipped with a DDR memory controller module. It is also necessary to organize a multi-buffer storage of the current frames of video sequence in DDR memory. If these requirements are met, the core can be connected with the FPGA project. The core being added, its interface should be linked with the FPGA project. Before starting the operation and every time the core is powered up, it is necessary to configure the device (write configuration registers). Once the configuration registers are written, it is possible to carry out control of tracking. The core does not provide a display of information. The display should be provided by the programmer. A detailed description of the core interface and the procedure of use are given in Parts 2 and 3 of this document.

PART 2 CORE SPECIFICATIONS

PERFOMANCE

The IP core characteristics given in Table 2 have been calculated from the results of tests performed on FPGA Xilinx Spartan-6XC6SLX150T-3. The performance characteristics depend significantly on the type of FPGA and image refresh rate. Table 3 shows throughput characteristics of the core and their specific features.

Table 3 – Basic characteristics of the D-MOT-10 IP core and their specific features

Parameter	Values and notes
Detecting mode	Automatic detection of moving objects with the determination and prediction of trajectories of objects based on the analysis of the sequence of frames in the video stream.
Maximum number of objects detected simultaneously	The core provides detection of up to 128 simultaneously moving objects on the frames in the video stream. Unique number is assigned to each detected object. Range from numbers assigned is 0 to 127 incl. The number kept on the object upon it is lost from sight. Herein the constant prediction of its trajectory is performed. The number is kept on the object even after its short-time disappearing behind an obstacle. When the object gets behind an obstacle the core continues to give the information about the object with the predicted coordinates. If the object does not appear again after 150 cycles (150 processed images) it is removed from the detection and its number is released and can be assigned to another object. The numbering of object is performed cyclically. This means when the object is removed from detection its number can be assigned to the newly detected object. A developer should make an assumption that while the information about a particular object number is given for each image frame this means that the object with this number presents in the image. If the delivery of the information about a particular number is interrupted for at least one processing cycle (1 frame in video sequence) than the objects corresponding to the number before and after the interruption can be different objects.
Maximum and minimum sizes of detecting strobes	The detection strobe size depends only on the size of the object and can’t be set by a command. The strobe size for an object can be changed from frame to frame depending on the changes in the shape and the size of the object.
Maximum size of detecting object	The maximum size of detected object is limited by the size of processed area (the size of the frame buffer).
Minimum size of detecting object	The core ensures stable detecting of moving objects of up to 8x8 pixels size in an image area. The smaller size of objects in the image plane is typical for local items such as tree branches, etc. These objects are designated as inessential and ignored by the core algorithms reducing the possibility of false detections.
Maximum and minimum movement speed of detecting object	For a sudden change of the position of tracking object, the speed is no more than 30 pixels per frame in any direction and no less than 0.1 pixels per frame in any direction. When detecting objects the core algorithms estimates the trajectory of motion and decline the objects with random trajectory. For the effective detection of moving objects it is important for the cameras field of view to be fixed (fluctuation no more than 2-3 pixels in the image plane).

Parameter	Values and notes
Minimum contrast of tracking objects	The core ensures stable tracking of objects featuring up to 10% contrast. This feature depends significantly on the characteristics of the background on which the object is observed. In the case of uniform and stationary background the contrast required for sustainable detection of the object can be significantly lower than the figure stated above.
Maximum size of frames	The core provides the work with frame sizes up to 720x576 pixels. Reading of images from DDR memory is carried out at specified starting memory addresses which are written in the core configuration registers. If it is necessary to use the core for video sequences with large frame sizes, the core can be synthesized to meet the required parameters (up to 5120x5120 pixels). In this case, the FPGA resources required for the core will have slightly higher parameters. A designer can use several cores in parallel to process large-sized images.
Elongation of tracking objects	Tracking objects can have any configuration.
Output coordinates accuracy	Output coordinates of tracking objects are obtained with 1 pixel accuracy. The output information points to the position of the tracking objects in the screen coordinate system (relative to top left corner of the image).
Time lag of output coordinates	At the frame refresh rate of 25 Hz and recommended sizes of the tracking strobe, the core calculates the object coordinates before the end of the next frame in the video sequence (delay is 1 frame). For other FPGA types, a higher processing speed is possible, and there is a possibility of using video sources with a higher frame-repetition rate.
Time from the core initialization to stable detecting mode	After the core initialization it needs 2-3 minutes to adapt to the current conditions of observation, provided immobility of the camera view. At this time the core initialize its filters and defines the image areas where constant random motion within a small range is observed (in the image plane) to exclusion them from further analysis in order to reduce the possibility of false detections.
Format of processed images	The core works with images having color palette of 256 colors (grayscale) – 8 bits/pixel.
Functioning conditions	The observed scene contained in the image sequence should be fixed (the field of view is stable and does not move in time). Slight fluctuation in the range of 2-3 pixels is possible; this fluctuation can be caused by the jitter of the video source under the influence of external factors.

REQUIRED RESOURCES

Table 4 shows the resources required for the D-MOT-10 IP core for various FPGA's of Xilinxi company.

Table 4 – Resources required for the D-MOT-10 IP core

FPGA type	Slice Registers	DSP	Block Memory
Artix-7 XC7A100T-3	5114 (4%)	11 (4%)	40 (29%)
Kintex-7 XC7K410T-2L	5113 (<1%)	11 (1%)	40 (5%)
Virtex-7 XC7VX330T-2L	5113 (<1%)	11 (<1%)	40 (5%)
Virtex-7 XC7V2000T-2L	5113 (<1%)	11 (<1%)	40 (3%)
Virtex-6 XC6VLX240T-2	5116 (1%)	11 (1%)	40 (9%)
Spartan-6 XC6SLX150T-3	5206 (2%)	13 (7%)	31 (11%)

Zenq-7 XC7Z045-2	5113 (<1%)	11 (<1%)	40 (7%)

Note: data can change depending on project characteristics.

REQUIREMENTS FOR FGPA PROJECT

The use of D-MOT-10 IP cores does not impose any substantial requirements on the structure of FPGA project. The designer needs to ensure connection of the core with DDR memory controller, configuration of registers, timing, control, and reception of information from the core. In addition, the designer needs to ensure refreshment of video frame in the DDR memory based on multi-buffering principle. Figure 1 shows an example of the project FPGA structure using the D-MOT-10 IP core.

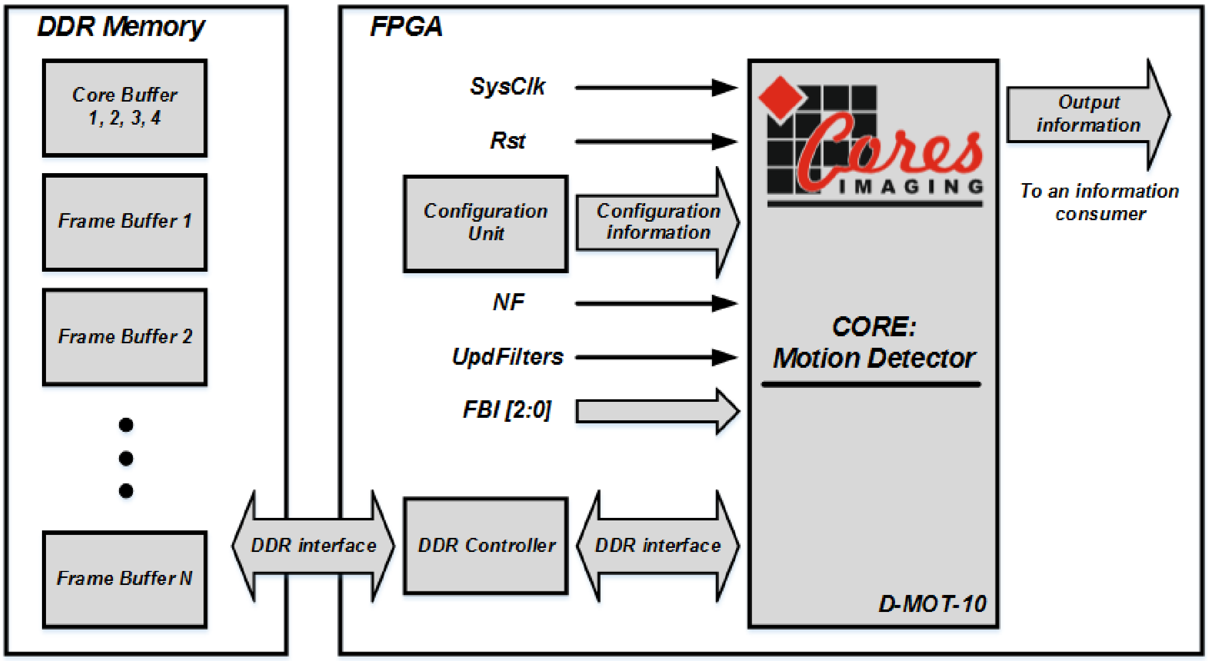

Figure 1 – An example of FPGA project structure using the D-MOT-10 IP core

The figure shows the interaction of the IP core with FPGA project. The core is connected to the DDR memory controller via a unified interface. The DDR memory controller can be either a hardware controller or the one synthesized by the core. Some FPGA Xilinx's have a hardware memory controller DDR. By this interface the D-MOT-10 core communicates with the DDR memory. The designer has to organize storage of the current frame sequence in the DDR memory. Storage of information should be organized in several buffers. The principle of video data storage can be explained as follows. Suppose there are N frame storage buffers as shown in Figure 1. The current image frame is already written in the buffer 1. At this time, the next frame is being written to the buffer 2. Starting addresses of each of the buffers are written in the configuration registers of the IP core. The index of the buffer where the image is stored for processing is transmitted to the core via data bus FBI [2:0] (index 0 - the first buffer). In this case, the core reads the current frame from the buffer 1. When a new frame is written in the buffer 2, the indices are changed so that a new image has to be written to the buffer 1 (index 0). At the same time, the value 1 is fed to the core via data bus FBI [2:0] which means that reading of the frame for processing has to be done from the buffer 2. It is assumed that the core has completed processing of the frame by the time a new frame arrives. If the core has not had time to process the previous frame, it will continue processing of the new one (buffer 2).

A set of buffers is required to arrange operation of several subsystems such as motion detection core, video recording module, a module for transfer of video information to the user, etc. The designer has to properly organize the changing of the frame buffer indices. The IP core reads the next frame in video sequence at the buffer address written in the configuration register. Configuration of the registers is carried out by a special module (Configuration Unit) before the work is started. This module is developed independently and it delivers the information sequence required to configure the core. The designer can use a ready-made module provided with the core or develop it himself. Writing of information to the configuration registers makes it possible to configure the appropriate core parameters before starting work. The designer must also provide synchronization and the core reset signal as well as the new frame signal (NF). The NF signal warns the core that the current image is written and its processing can be started. The designer organizes control of the core working process by sending control messages (Control Information) via the control interface. This interface is used to receive information from the control module about the capture or rejection of tracking object, change of tracking parameters, etc. The output information is sent to the user via a unified interface. This information can be delivered to the executive subsystem, display subsystem, directly to communication link, etc. The core also needs some space in the DDR memory (Core Buffer). It is recommended to allocate no less than 82 Kbytes for such memory space. The size of the Core Buffer changes dynamically depending on the operating mode. The start address of this memory space is written to the configuration registers before the work is started. The interface for data exchange with the D-MOT-10 IP core is described in the next section.

DESCRIPTION OF INTERFACE

Figure 2 shows the interface for interaction with the D-MOT-10 IP core.

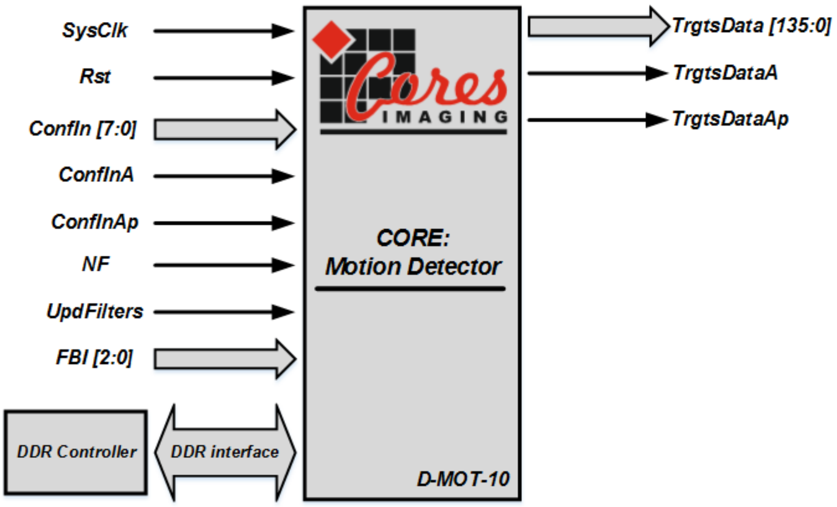

Figure 2 – Interface for interaction with the D-MOT-10 IP Core

All input and output assignments are given in Table 5.

Table 5 – Assignment of the inputs and outputs of the D-MOT-10 IP core

№	Name of signal	Name of signal
1	*SysClk*	Clock signal for the core. All signals for the core (control, configuration, configuration, reset, etc.) must be synchronized with SysClk. When testing for the FPGA Xilinx Spartan-6150-based project, SysClk frequency was chosen to be 130 MHz. The optimal clock signal frequency should be determined by the designer for a specific project.
2	*Rst*	Core reset signal. The signal must be synchronous with SysClk and have duration of 1 clock cycle. The reset signal is used 1 time at the beginning of work of the core to write the configuration registers.

№	Name of signal	Name of signal
3	*ConfIn [7:0]* *ConfInA* *ConfInAp*	Interface for writing configuration registers. By this interface, the configuration registers are written before the work is started. Values of the configuration registers are passed via the ConfIn [7:0] bus. The ConfInA signal warns about data activity (data writing). ConfInAp signals the use of data in the registers. The interface is a unified one. Timing diagrams of the information exchange are given in Figure 3.
4	*NF*	New frame signal of 1 clock cycle duration. The signal must always come when video frame changes. Upon this signal, the core starts processing of the next frame.
5	*FBI [2:0]*	Bus for the current frame buffer index. This bus sets the buffer index of the frame which needs to be processed. Note that the addresses of each of the buffers are written to the memory via the interface for configuration registers writing. The core reads a particular buffer from the DDR memory in accordance with the bus value.
6	*UpdFilters*	The core filters reset signal. Is applied in case it is necessary to re-start the adaptation to the changed conditions of observation (e.g. changes in the observed scene by the rotation of the camera).
7	DDR Interface	Unified interface for data exchange with the DDR Xilinx memory controller. The interface is shown in Figure 4.
8	*TrgtsData [135:0]* *TrgtsDataA* *TrgtsDataAp*	Interface for the output of information about the coordinates of detecting objects. This interface transmits the coordinates of detecting objects. The interface is unified with the control interface and configuration-register-writing interface. Timing diagrams of the information exchange are presented in Figure 3.

Figure 3 shows the timing diagrams for the exchange of information via the data-output, control and register writing interfaces. The interfaces are identical in terms of information exchange principle used.

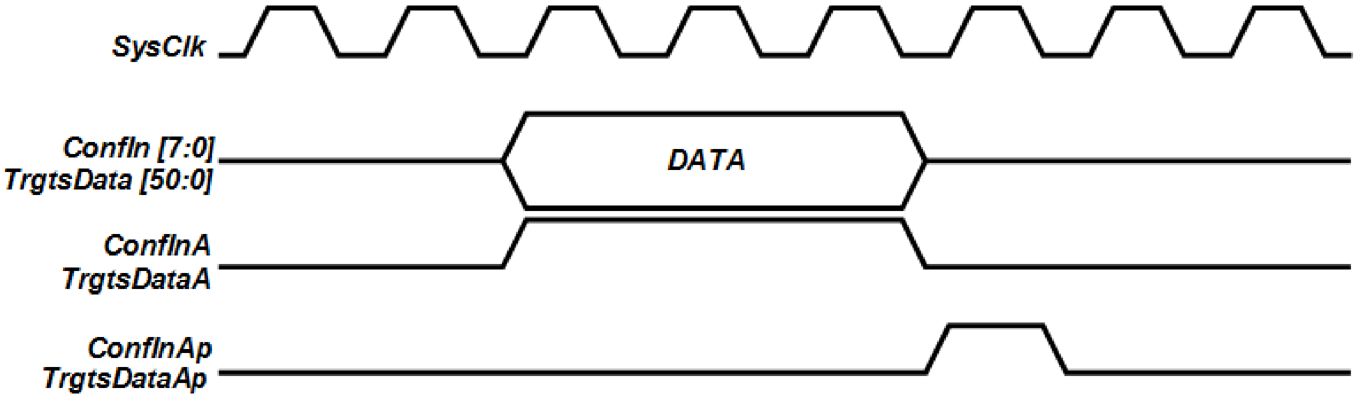

Figure 3 – Timing diagrams for the exchange of information via the coordinate data output, control and configuration register writing interfaces

All data exchange interfaces are synchronous with the SysClk signal. The Interface for data exchange with the DDR memory controller is unified for all FPGA Xilinx's. A detailed description of the DDR controller interface is given in the DDR controller interface shown in the documentation on the DDR memory controller core from Xilinx ®. Figure 4 shows the picture of the interface to the DDR memory controller from the D-MOT-10 IP core.

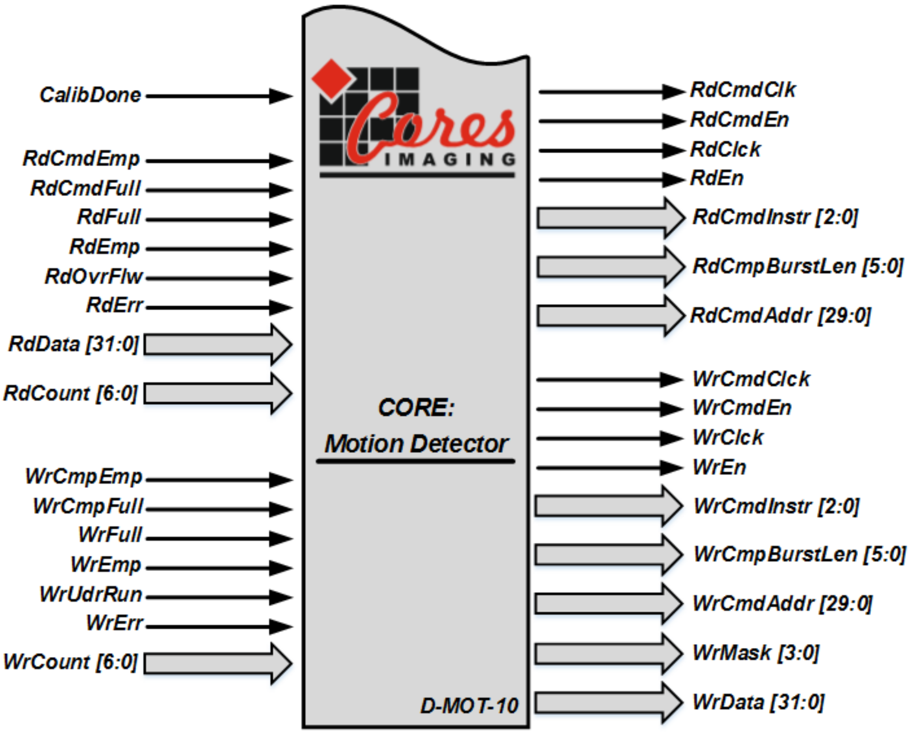

Figure 4 – Interface to the DDR memory controller

Assignments of the inputs and outputs of the interface to DDR memory controller is presented in Table 6 along with their correspondence with inputs and ouputs given in Xilinx specifications.

Table 6 – Input and output assignments of the D-MOT-10 IP core

№	Name of T-COR-10 signal	Direction	Memory controller signal	Assignment
1	*CalibDone*	input	*calibe_done*	Signal for end of calibration of DDR memory controller.
2	*RdCmdEmp*	input	*pX_cmd_empty*	Read command queue is empty.
3	*RdCmdFull*	input	*pX_cmd_full*	Read command queue is full.
4	*RdFull*	input	*pX_rd_full*	Read buffer is full.
5	*RdEmp*	input	*pX_rd_empty*	Read buffer is empty.
6	*RdOvrFlw*	input	*pX_rd_overflow*	Read buffer overflow.
7	*RdErr*	input	*pX_rd_error*	Read error.
8	*RdData [31:0]*	input	*pX_rd_data [PX_SIZE-1:0]*	Read data bus.
9	*RdCount [6:0]*	input	*pX_rd_count [6:0]*	Bytes read.
10	*WrCmdEmp*	input	*pX_cmd_empty*	Write command queue is empty.
11	*WrCmdFull*	input	*pX_cmd_full*	Read command queue is full.
12	*WrFull*	input	*Px_wr_full*	Write buffer is full.
13	*WrEmp*	input	*pX_wr_empty*	Write buffer is empty.
14	*WrUdrRun*	input	*pX_wr_underrun*	No space in FIFO for new data.
15	*WrErr*	input	*pX_wr_error*	Write error.
16	*WrCount [6:0]*	input	*pX_wr_count [6:0]*	Bytes written.
17	*RdCmdClck*	output	*pX_cmd_clk*	Read command clock rate.
18	*RdCmdEn*	output	*pX_cmd_en*	Read command strobe.
19	*RdClck*	output	*pX_rd_clk*	Data read clock rate.
20	*RdEn*	output	*Px_rd_en*	Data read strobe.
21	*RdCmdInstr [2:0]*	output	*pX_cmd_istr [2:0]*	Read command type.

№	Name of T-COR-10 signal	Direction	Memory controller signal	Assignment
22	*RdCmdBurstLen [5:0]*	output	*pX_cmd_bl [5:0]*	Read message length.
23	*RdCmdAddr [29:0]*	output	*pX_cmd_addr [29:0]*	Read address.
24	*WrCmdClck*	output	*pX_cmd_clk*	Write command clock rate.
25	*WrCmdEn*	output	*pX_cmd_en*	Write command strobe.
26	*WrClck*	output	*pX_wr_clk*	Data write clock rate.
27	*WrEn*	output	*pX_wr_en*	Write data strobe.
28	*WrCmdInstr [2:0]*	output	*pX_cmd_instr [2:0]*	Write command type.
29	*WrCmdBurstLen [5:0]*	output	*pX_cmd_bl [5:0]*	Write message length.
30	*WrCmdAddr [29:0]*	output	*pX_cmd_addr [29:0]*	Write address.
31	*WrMask [3:0]*	output	*pX_wr_mask [PX_MASKSIZE-1:0]*	Writing mask.
32	*WrData [31:0]*	output	*pX_wr_data [PX_SIZE-1:0]*	Write data bus.

Notes:

1. Prefix “pX” in the name of signals of the DDR memory controller denotes port number. In this, “p” – acronym of “port”, and “X” will be replaced by number in a specific project.
2. In the table, some of the DDR memory controller signals are repeated, but they will specify contacts with different numbers. The reason for this is that the D-MOT-10 IP core exchange interface uses 2 identical controller ports (configured for reading and writing). General connection layout is presented in Figure 5.

The designer does need to know the procedure of data exchange with the DDR controller. To do this, it is necessary to generate the DDR controller core with two ports (for reading and writing) in the development environment (XilinxISE of required version) and, then, connect the corresponding groups of signals of the core and DDR controller. Figure 5 gives a schematic view of the correspondence between the signal names and reading and writing ports of the DDR controller.

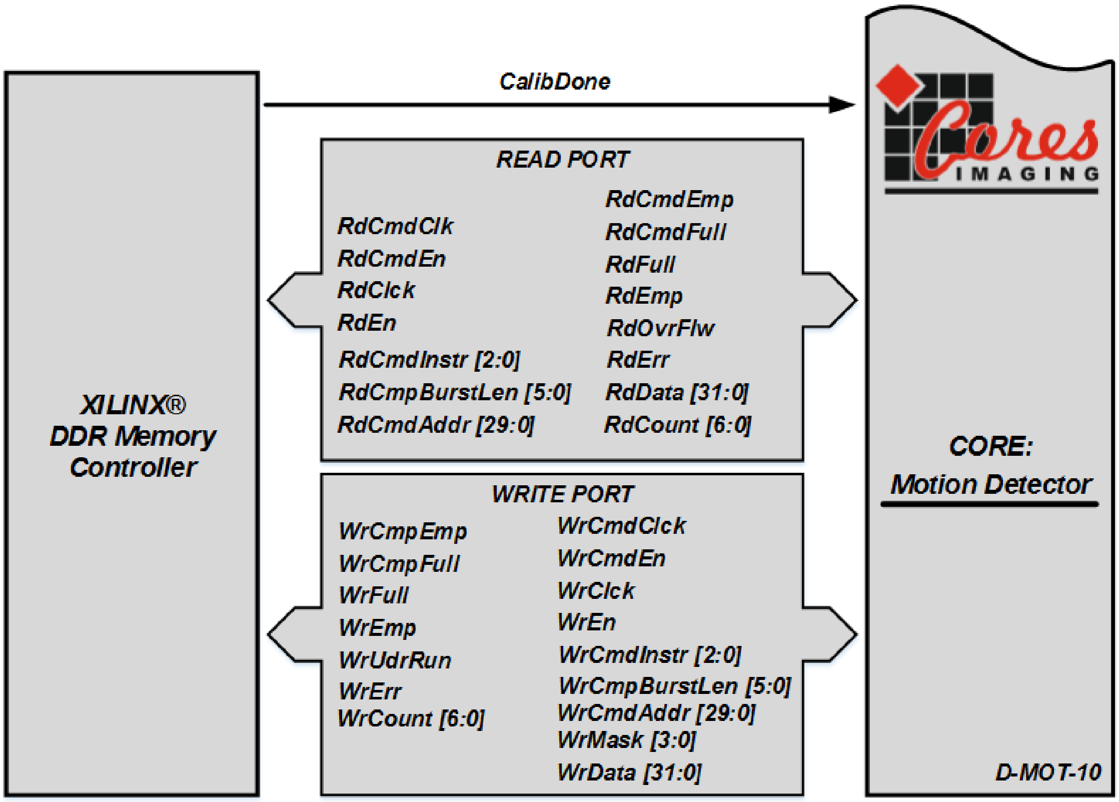

Figure 5 – Correspondence of the input/outputs of the M-MOT-10 IP core and the reading and writing ports of the DDR memory controller

As seen from the figure, to ensure data exchange of the D-MOT-10 IP core with the DDR controller, the latter must be configured with two ports (read and write) and with the CalibDone signal to the core. The incorporation of the D-MOT-10 IP core into an FPGA project is described in detail in Part 3.

CONFIGURING THE CORE

Before the work is started, the D-MOT-10 IP core should be configured. Configuring consists in the writing of the configuration registers. The user is given an option to carry out parameter setting of the core algorithms by himself. Configuration is performed through the unified interface for writing configuration registers as described in the previous section. This interface works with the signals: ConfIn [7:0], ConfInA, ConfInAp. The IP core parameters available for setting are shown in Table 6.

Table 6 – Parameters of the IP core available for setting

Parameter name	Designation	Value	Assignment
Frame buffer address 1	FB1	4 bytes of memory address	Represents address of the first memory cell of frame buffer 1.
Frame buffer address 2	FB2	4 bytes of memory address	Represents address of the first memory cell of frame buffer 2.
Frame buffer address 3	FB3	4 address bytes	Represents address of the first memory cell of frame buffer 3.

Parameter name	Designation	Value	Assignment
Frame buffer address 4	FB4	4 bytes of memory address	Represents address of the first memory cell of frame buffer 4.
Frame buffer address 5	FB5	4 address bytes	Represents address of the first memory cell of frame buffer 5.
Service area address 1 (CoreBuffer 1)	CB1	4 bytes of memory address	Represents address of the first memory cell of the core buffer. The size of this buffer is equal image width * image height (width and height in bytes).
Service area address 2 (CoreBuffer 1)	CB2	4 bytes of memory address	Represents address of the first memory cell of the core buffer. This buffer stores the equivalent for the amount of movements for each point in the frame (selection of moving objects in the frame). The designer can read the data from the frame and use them to display moving objects (useful for detecting moving objects visually). The size of this buffer image is equal width * image height (width and height in bytes).
Service area address 3 (CoreBuffer 1)	CB3	4 bytes of memory address	Represents address of the first memory cell of the core buffer. The size of this buffer is equal 2 * image width * image height (width and height in bytes).
Service area address 4 (CoreBuffer 1)	CB4	4 bytes of memory address	Represents address of the first memory cell of the core buffer. The size of this buffer is equal 0,125 * image width * image height (width and height in bytes).
Frame buffer width	FBW	2 bytes (permitted values from 8 to 2047)	Frame buffer width in pixels (width of images arriving from video source).
Frame buffer height	FBH	2 bytes (permitted values from 8 to 2047)	Frame buffer height in pixels (height of images arriving from video source).
Detection threshold	P	1 byte (permitted values from 0 to 255)	This parameter defines the detection threshold from 0 to 255 which is equivalent to a range from 0 to 1.

To configure the core, it is necessary to transmit 45 bytes of the configuration data (configuration packet) via the corresponding interface before the work is started (at the time of power-up). After this, the core is ready for operation. If there is a need to change the parameters in the process of work (e.g., change smoothing factor or the width and height of the region under check), it is possible to re-write the parameters promptly during tracking without turning off or resetting the core. Table 7 shows the sequence of configuration data in the configuration packet.

Table 7 – The sequence of the configuration data

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Data	FB1				FB2				FB3				FB4				
Byte	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Data	FB5			CB1				CB2				CB3				CB4	
Byte	34	35	36	37	38	39	40	41	42	43	44						
Data	CB4		CB5				FBW		FBH		P						

Explanations with respect to each of the parameters are given below.

Frame buffer addresses. Before the work is started, it is necessary to specify addresses of the first memory cells of the frame buffers. Setting of up to 5 frame buffers is possible. At a time, the core works with that frame buffer whose number has been transmitted to the core via the FBI [2:0] bus. Addresses of the first cells of frame buffers can be changed during operation, which allows the designer to use more of them. The IP core works with images in 8 bits/pixel format (grayscale), so the designer has to allocate memory in a project bearing this in mind. It is also necessary to meet the requirement that addresses of all memory spaces (frame buffers and core buffer) are multiples of 4.

Core buffer address. The service area of memory is used by the core to store the current settings, intermediate results of work and image fragments. The size of each core buffer differs from the others (CB1, CB2, CB3, CB4, CB5). To calculate the memory size the following correlations are applied:

CB1 = FDW*FBH (bytes);
CB2 = FDW*FBH*2 (bytes);
CB3 = FBW*FBH (bytes);
CB4 = FBW*FBH*2 (bytes);
CB5 = (FBW*FBH)/8 (bytes).

Frame buffer width. This width corresponds to the width of images under processing. The frame buffer width is measured in bytes, and with the format of images of 8 bits/pixel corresponds to their width. In addition, these values are used by the core to restrict the movement of tracking strobes within the image boundaries (the strobe cannot go beyond the image boundaries. This parameter can be changed by user in the process of work.

Frame buffer height has the same physical sense as the frame buffer width.

Detection threshold. This parameter changes in the range from 0 to 255 which is equivalent to the threshold probability of object detection in the image from 0 to 1.

If it turns out that in the course of tracking the probabilities of finding the object in all search positions (search area) are lower than the threshold value, the core switches to the path prolongation mode (extension of the tracking object path based on the calculated parameters of its movement) until the probability of finding object in a given point is above the threshold (automatic re-capture of the object for tracking).

If it is necessary to change the core parameters during work, the whole configuration packet should be transmitted together with all changes required. In order to configure the core more conveniently, it is recommended to synthesize a special module (core) which will transmit the required messages to the D-MOT-10 IP core. If the core parameters need to be changed quickly and efficiently, this function should be assigned to the control module of the whole FPGA project.

CONTROL OF OPERATION MODES

IP Core D-MOT-10 works in one operation mode – detection of moving objects. The designer can control only operating parameters while the core is working. If nesessary the developer can manage the core filters upgrade (reset the filters by *UpdFilters signal*).

OUTPUT INFORMATION

The output information from the IP core D-MOT-10 is available via a unified interface for receiving data from the core (OutTrgtData [7:0], OutTrgtDataA, OutTrgtDataAp). The data exchange principle is analogous to that used in the configuration-register-writing interface and control interface, the only difference being in the data transmission direction. The core provides information for each detected object separately. The packet contains the number assigned to the object, detecting strobe corners coordinates and motion direction. The output information represents packets 17 bytes long. The format of the output data packets is presented in Table 10.

Table 10 – Format of the output information packets

Bit	135	134	133	132	131	130	129	128	127	126	125	124	123	122	121	120
Data	N								X1							
Bit	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104
Data	X1								Y1							
Bit	103	102	101	100	99	98	97	96	95	94	93	92	91	90	89	88
Data	Y1								X2							
Bit	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72
Data	X2								Y2							
Bit	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56
Data	Y2								Vx							
Bit	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40
Data	Vx															
Bit	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24
Data	Vx								Vy							
Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Data	Vy															
Bit	7	6	5	4	3	2	1	0								
Data	Vy															

Table 11 presents types of data and their assignments.

Table 11 – data types and their assignments

Data type	Value	Assignment
N	8	Serial number of detected object. The core assigns a unique number to each detected object. The number is assigned cyclically. If the object with a particular number (e.g. number 1) is lost from the field of view and is not detected again, its number can be assigned to a new object. Maximum number of the objects detecting simultaneously is 32.
X1	16	The coordinate of the left upper corner of the detecting strobe in window coordinate system, as shown in Figure 6.
Y1	16	The coordinate of the left upper corner of the detecting strobe in window coordinate system, as shown in Figure 6.
X2	16	The coordinate of the right lower corner of the detecting strobe in window coordinate system, as shown in Figure 6.

Y2	16	The coordinate of the right lower corner of the detecting strobe in window coordinate system, as shown in Figure 6.
Vx	32 (the highest bit is significant)	The component of the path velocity in window coordinate system axis X, as shown in Figure 6. The highest bit is significant (1 means a negative number). Vx value is represented in relative units and does not correspond to image pixels.
Vy	32 (the highest bit is significant)	The component of the path velocity in window coordinate system axis Y, as shown in Figure 6. The highest bit is significant (1 means a negative number). Vy value is represented in relative units and does not correspond to image pixels.

Information about each object is given separately. Herein all the information is passed while a new frame is being processed. If the object moves beyond the edge of the image (left, top, bottom or right edge of the detecting strobe moves outside the image), the core leaves the coordinates of the corners of the detecting strobe unchanged but modifies the other coordinates. Herein when the object moves outside the image the output information will contain constantly decreasing detecting strobe size of the object until the object disappears and is removed from detection. Figure 6 shows the image of window coordinate system with the explanation to the output information.

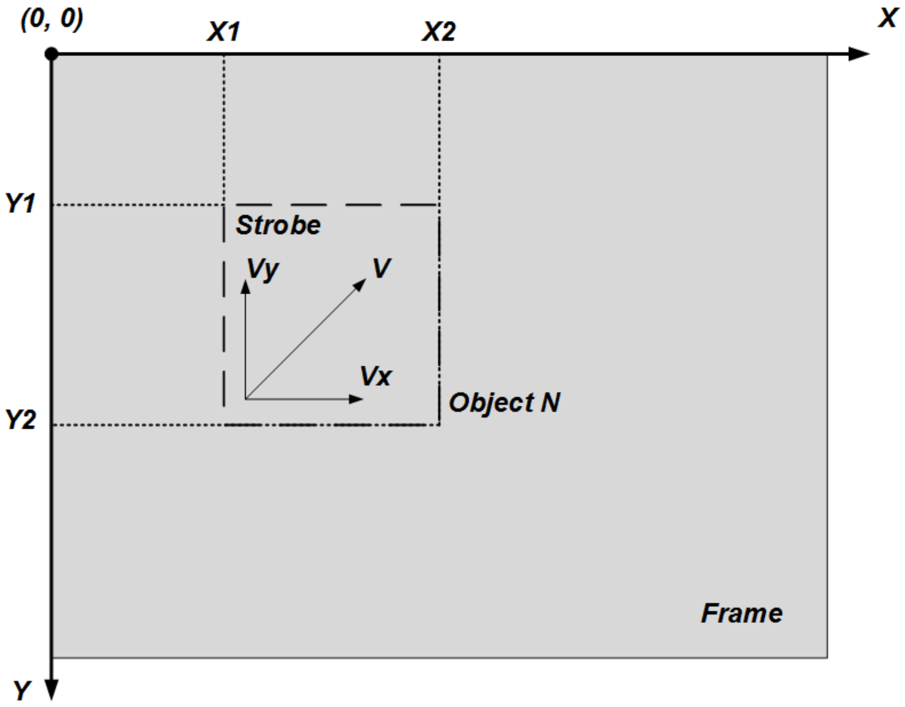

Figure 6 – Explanation to the output information of the D-MOT-10 IP Core

PART 3 WORKING WITH THE CORE

CONNECTING THE CORE TO FPGA PROJECT

Since the core is supplied for the FPGA Xilinx project, the procedure of connection of the D-MOT-10 IP core is presented for IDEISE 14.3. The core represents a file with extension *. ngc, a file with extension *.sym and a file with extension *.v (On request, we can supply *. Vhdl) (D_MOT_10_XXXXXX.ngc, D_MOT_10_XXXXXX.sym, D_MOT_10_XXXXXX.v (. Vhdl)). The symbols "X" in the file name stand for the FPGA model name for which the core is synthesized. The file D_MOT_10_XXXXXX.ngc is the synthesized core. The file D_MOT_10_XXXXXX.sym is graphic symbol of the core. The file D_MOT_10_XXXXXX.v (. Vhdl) is the shell where the core inputs and outputs are specified. Below is the procedure of using the core in an FPGA project with a Schematic upper level in IDE Xilinx ISE 14.3.

1. Copy the file D_MOT_10_XXXXXX.sym to your project folder.

2. Add the file D_MOT_10_XXXXXX.v (.vhdl) to the project.

3. Place the graphic symbol of the core in the upper module. This is done as follows:

3.1.Select «Symbols» tab.

3.2.Select 'MotionProcessor' in the 'Symbols' list.

3.3. Drag 'MotionProcessor' symbol to the main field of the document and position the core graphic symbol.

4. Connect the inputs and outputs of the core with the inputs and outputs of the memory controller core and control core (the control interface, configuration-register-writing interface and data-receive interface are connected with the interfaces of the (core) control module).

5. In the project synthesis parameter «Cores Search Directories», specify the path to the folder containing the file D_MOT_10_XXXXXX.ngc.

To use the core in the projects where the upper module is in the form of an HDL-file, follow these steps:

1. Add the file D_MOT_10_XXXXXX.v (.vhdl) to the FPGA project.

2. Mark the file added and run «View HDL Instantiation Template» in the section 'Processes'.

3. Copy the core template from the open window to the upper module.

4. Then, run items 4 and 5 in the Section for the use in projects with a Schematic-type upper module.

To simplify the integration of the core into finished projects, we supply project files together with Verilog file of the core configuration module TPConfig.v, which is linked with the definition file 'Definitions.v.' The configuration module is connected with the configuration-register-writing interface and makes it possible to configure the core with appropriate parameters before the work is started. The designer can set the required parameters in the file before he starts to synthesize the project. It is recommended to organize the initial configuration using a single control module which allows changing of settings in the process of work. The configuration module can be directly connected to the core 'MotionProcessor' via the configuration-register-writing interface.

CONTACTS

RIFTEK Sensors & Instruments

RIFTEK – a private company dealing with development and production of optoelectronic devices intended for measuring geometric quantities and video processing tools.

Logoiski Trakt 22-311, 220090 Minsk, Republic of Belarus

Tel./fax: +375 17 281-35-13; +375 17 281-36-57
GSM: +375 29 655-72-55
e-mail: info@riftek.com; sales@riftek.com
Website: www.riftek.com