IBFB packet player

ibfb_player
Revision 1.0

Firmware Data Sheet

PSI, 18.07.2016

Content

Table of Contents

1 Introduction

The IBFB system uses data from a set of BPMs in order to compute the corrections to be applied through the kickers. Since the number of available BPMs is high, it's not feasible to connect each of them to the IBFB controller. Instead a double daisy-chain topology is used: each BPM transmits data to the previous BPM in the chain (upstream link) and to the following (downstream link). In addition each BPM forwards to the upstream link the data coming from the downstream link. Likewise the data from the downstream link is forwarded to the upstream link. A simplified diagram of the BPM links is shown in Figure 1.

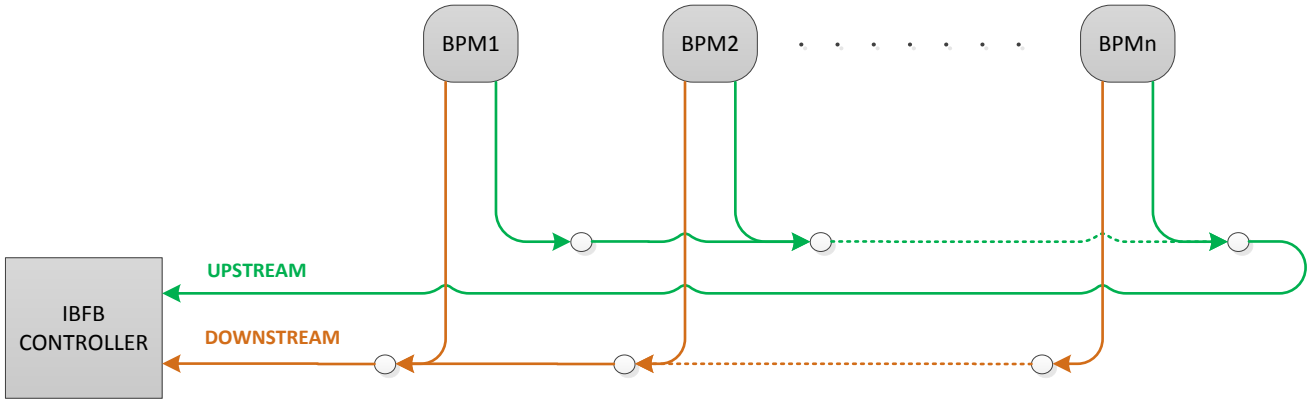

Figure 1 - Data link between BPMs (simplified)

Data from each BPM is formatted in packets according to a custom protocol (see §**Error! Reference source not found.** for details). In order to test the IBFB system during the development phase, a packet player has been developed. The player can emulate data transmission from an arbitrary number of BPMs. A RAM holds the packet data to be played-back. Each packet is transmitted with a specific delay with respect to an external trigger. The delay of each packet is also stored in the player's playback RAM.

1.1 Features

The *ibfb_player* component has the following features:

- PLB register interface for command and control (tested up to 125 MHz PLB clock frequency on Virtex-5). Each player channel has independent controls.
- Provides up to 10 independent player channels. Each channel uses a GTX serial transceiver for data transmission.
- Tested up to 3.125 Gbps GTX link speed on Virtex-5.
- Selectable core clock source (internal or external). Tested up to 216 MHz on Virtex-5.

1.2 Definitions, acronyms, and abbreviations

FPGA	Field Programmable Gate Array
PLB	Processor Local Bus (by IBM)
GPAC	Generic PSI ADC Carrier
BPM	Beam Position Monitor
IBFB	Intra-bunch Feedback

1.3 References

[1] ”Generic PSI ADC Carrier, VME64x FPGA Carrier Board for High Speed Mezzanines”, *Datasheet*, Board Rev. 2.1, 12.07.2013.
[2] PDC QSFP, Quad SFP Module, *Schematic*, Revision 0.11, 2.7.2013
[3] Virtex-5 FPGA RocketIO GTX Transceiver, User Guide, UG198, V3.0, 30.10.2009

1.4 History

Revision	Date	Author	Description
1.0	18.07.2016	A. Malatesta	First release

2 Firmware Description

The core is implemented in pure VHDL. It does contain device specific primitives (Xilinx's GTX Transceivers), therefore its portability is currently limited to Xilinx Virtex5 devices providing at least five GTX_DUAL tiles. The component is wrapped in a Xilinx EDK pcore with slave PLB interface, but its core logic (single player channel) is also available as an independent VHDL component.

2.1 Architecture

The core's architecture is shown in

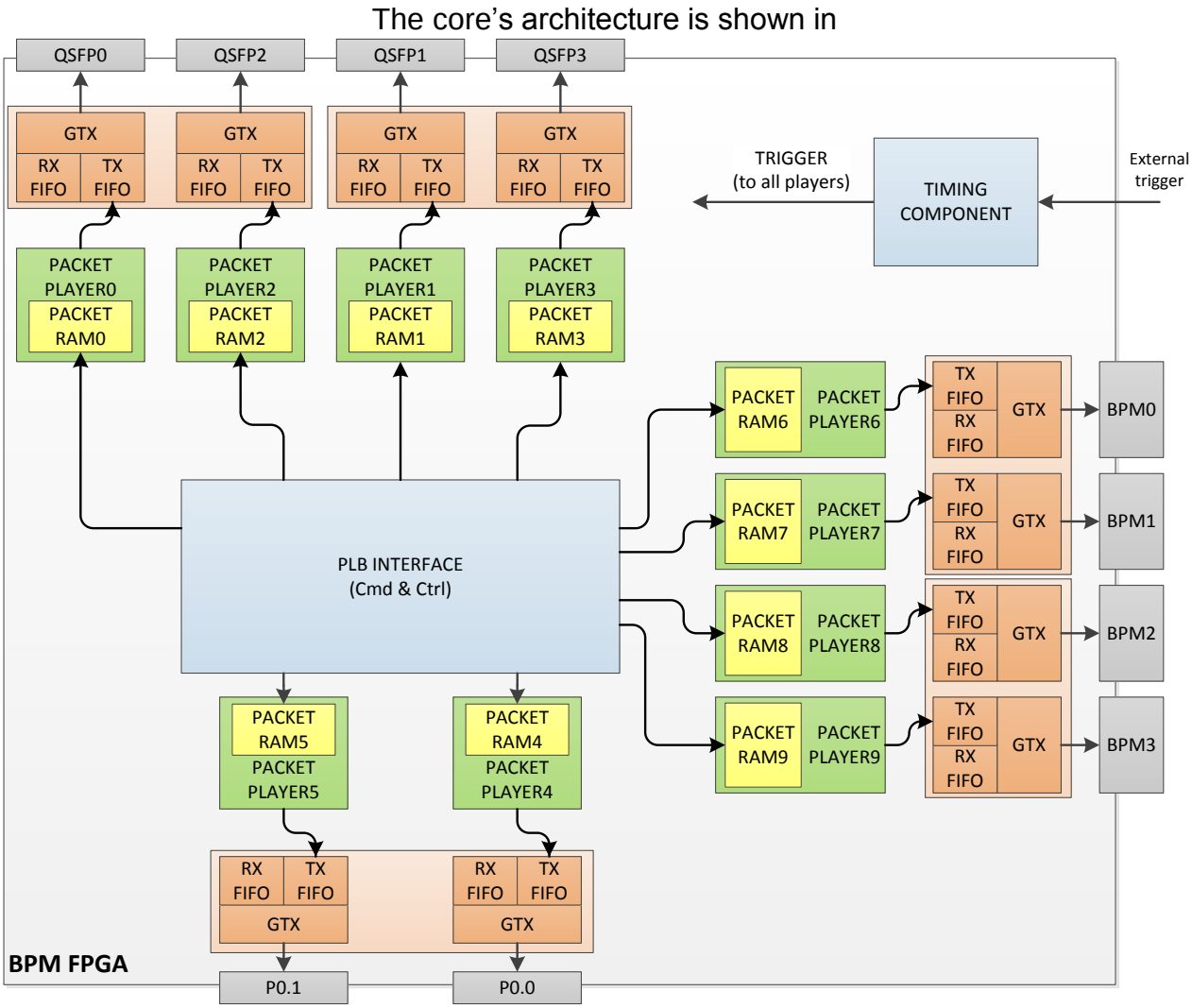

Figure 2. It is explicitly developed to be instantiated in the GPAC's BPM FPGA (cfr. [1]). Up to 5 channels pairs can be activated through via parameter. Each couple is connected to a GTX tile. Channels 0 to 3 are connected to a QSPF piggyback (cfr [2]). Channels 4 and 5 are connected to the backplane P0 connector. Channels 6 to 9 are connected to the other BPM FPGA on the same GPAC board. Each player is an independent component containing its own register bank and playback RAM.

A common PLB interface allows to access each channel and some common registers.

A timing component generates an internal trigger for all the players. This trigger is generated from an external one. An arbitrary trigger delay can be set via PLB interface.

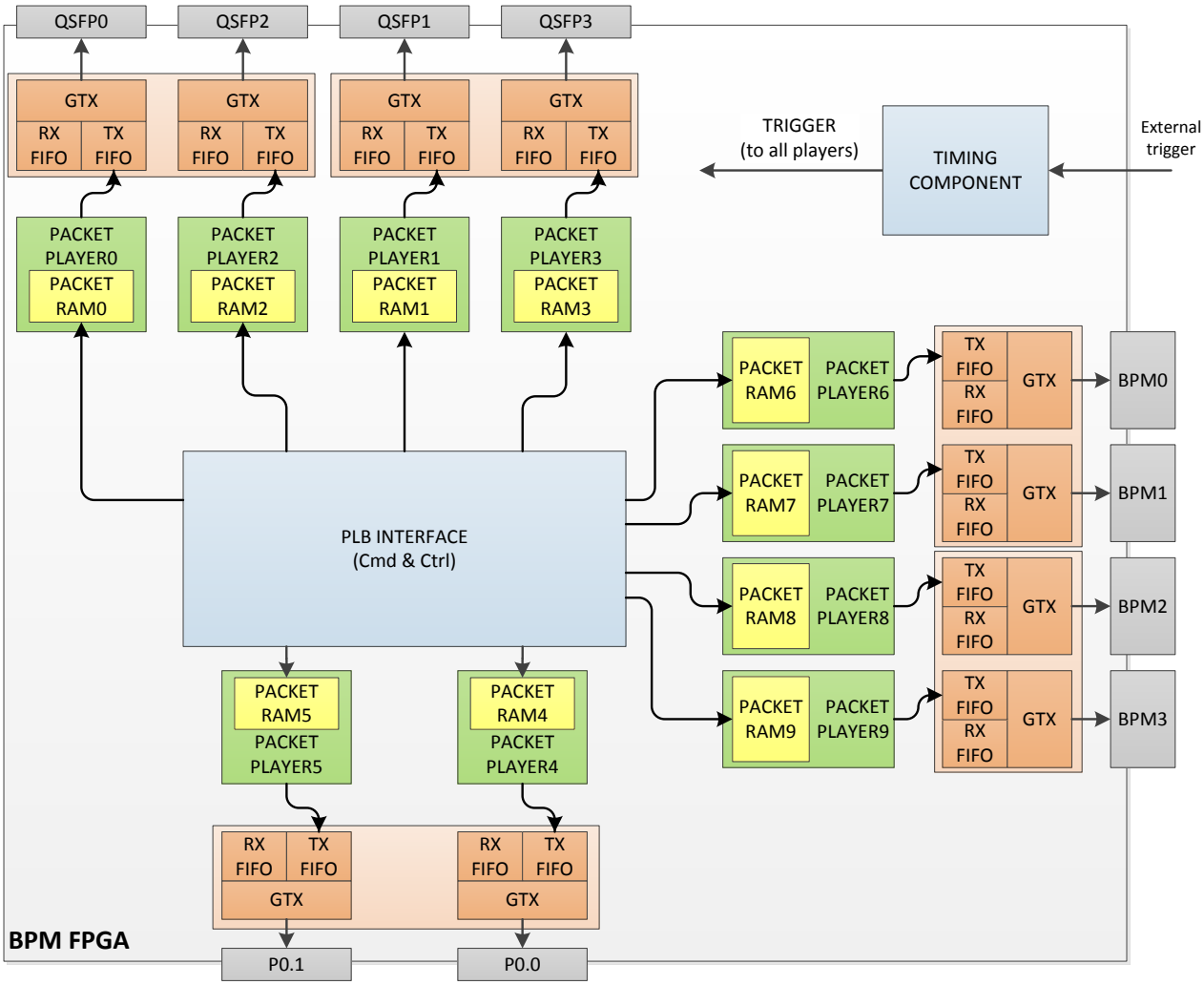

Figure 2 – Top-level block diagram of the core.

2.2 Ports and Attributes

Figure 3 contains component overview with input and output ports. Table 1 and Table 2 describe the meaning of port and attributes.

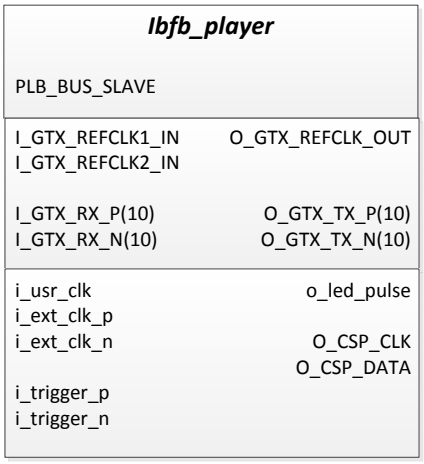

Figure 3 - Top entity of the ibfb_player.

Table 1 - Port description for ibfb_bpm_router

Port name	Dir.	Type	Description
I_GTX_REFCLK1_IN	in	std_logic	First reference clock for the GTX transceivers. Any of the five GTX tiles contained in the component can be assigned to use this clock by properly setting the generic C_GTX_REFCLK_SEL (cfr. Table 2).
I_GTX_REFCLK2_IN	in	std_logic	Second reference clock for the GTX transceivers. Any of the five GTX tiles contained in the component can be assigned to use this clock by properly setting the generic C_GTX_REFCLK_SEL (cfr. Table 2).
O_GTX_REFCLK_OUT	out	std_logic	NOT CONNECTED
I_GTX_RX_P	in	std_logic_vector(9:0)	Positive receive wires for the GTX physical layer differential pairs (see Table 3 for GTX channel assignment).
I_GTX_RX_N	in	std_logic_vector(9:0)	Negative receive wires for the GTX physical layer differential pairs (see Table 3 for GTX channel assignment).
O_GTX_TX_P	out	std_logic_vector(9:0)	Positive transmit wires for the GTX physical layer (see Table 3 for GTX channel assignment).
O_GTX_TX_P	out	std_logic_vector(9:0)	Negative transmit wires for the GTX physical layer (see Table 3 for GTX channel assignment).
i_user_clk	in	std_logic	Internal core clock. Used to drive logic only if the generic C_USE_EXTERNAL_CLOCK = '0'.
i_ext_clk_(p/n)	in	std_logic (differential pair)	External core clock. Used to drive logic only if the generic C_USE_EXTERNAL_CLOCK = '1'.
i_trigger_(p/n)	in	std_logic (differential pair)	External trigger. Fed to the timing component, and used to generate the start trigger for all the players.
o_led_pulse	out	std_logic	From the timing component. Meant to be connected to a LED in order to monitor trigger's activity.

Port name	Dir.	Type	Description
PLB_SLAVE	in	BUS	PLB slave bus (bursting) used to access local register bank for command and control purposes, and the players' RAMs. PLB bus clock is used to drive only the register bank logic and the RAM interface. The core itself runs with the clock specified by the C_USE_EXTERNAL_CLOCK parameter.

Table 2 - Attribute description for ibfb_bpm_router

Attribute Name	Type	Description
C_USE_EXTERNAL_CLOCK	std_logic	Select which clock to use for the players' core logic. When '0', then *i_user_clk* is used, otherwise *i_ext_clk* is used.
C_K_SOP	std_logic_vector(7:0)	K-character used as Start-of-packet symbol in IBFB protocol. Shall be a K-character accepted by Xilinx GTX transceivers 8b/10b logic (see [3]).
C_K_EOP	std_logic_vector(7:0)	K-character used as End-of-packet symbol in IBFB protocol. Shall be a K-character accepted by Xilinx GTX transceivers 8b/10b logic (see [3]).
C_PLAYER_EN	std_logic_vector(0:9)	Each bit enables one player: • Bit 0: QSFP channel 0 • Bit 1: QSFP channel 1 • Bit 2: QSFP channel 2 • Bit 3: QSFP channel 3 • Bit 4: P0 channel 0 • Bit 5: P0 channel 1 • Bit 6: BPM channel 0 • Bit 7: BPM channel 1 • Bit 8: BPM channel 2 • Bit 9: BPM channel 9
C_PLAYER_CTRL_EOS	std_logic_vector(7:0)	Player stops automatically whenever it reads from its RAM a packet for which the CTRL field is equal to the value specified by this parameter. See §2.4.1 for details on the packet fields.
C_GTX_REFCLK_SEL	std_logic_vector(4:0)	Define which reference clock is used by every GTX slice. Each bit represent a slice: • Bit 4: cross-FPGA channels 2 and 3 • Bit 3: cross-FPGA channels 0 and 1 • Bit 2: P0 channels 0 and 1. • Bit 1: QSFP channels 2 and 3 • Bit 0: QSFP channels 0 and 1 Value 0 selects I_GTX_REFCLK1_IN, while value 1 selects I_GTX_REFCLK2_IN. According to which physical transceivers are used, restrictions apply (see [3] for details). For configuration described in Table 3, the value MUST be "01100".
C_PLAYER_RAM_ADDR_W	natural	Address width for the players' playback RAM. Let this value be log2(4*N+2) where N is the maximum number of packets to be stored in each player. This leaves room for a spare packet containing the END-OF-STREAM field (see §2.4 for details).
C_SFP02_REFCLK_FREQ	natural	Reference clock frequency for QSFP channels 0 and 2 (in Megahertz)
C_SFP13_REFCLK_FREQ	natural	Reference clock frequency for QSFP channels 1 and 3 (in Megahertz)
C_P0_REFCLK_FREQ	natural	Reference clock frequency for P0 channels 0 and 1 (in Megahertz)
C_BPM_REFCLK_FREQ	natural	Reference clock frequency for cross-FPGA channels (in

		Megahertz)
C_SFP02_BAUD_RATE	natural	Baud rate for SFP channels 0 and 2 (in Kilobits per second)
C_SFP13_BAUD_RATE	natural	Baud rate for SFP channels 1 and 3 (in Kilobits per second)
C_P0_BAUD_RATE	natural	Baud rate for P0 channels 0 and 1 (in Kilobits per second)
C_BPM_BAUD_RATE	natural	Baud rate for cross-FPGA channels (in Kilobits per second)

Table 3 - GTX channels assignment.

GTX channel index	Signals	Description
GTX SFP Channel 0	I_GTX_RX_P(0) I_GTX_RX_N(0) O_GTX_TX_P(0) O_GTX_TX_N(0)	Connected to GPAC's PB_8MGT_*(4) signal.
GTX SFP Channel 1	I_GTX_RX_P(1) I_GTX_RX_N(1) O_GTX_TX_P(1) O_GTX_TX_N(1)	Connected to GPAC's PB_8MGT_*(6) signal.
GTX SFP Channel 2	I_GTX_RX_P(2) I_GTX_RX_N(2) O_GTX_TX_P(2) O_GTX_TX_N(2)	Connected to GPAC's PB_8MGT_*(5) signal.
GTX SFP Channel 3	I_GTX_RX_P(3) I_GTX_RX_N(3) O_GTX_TX_P(3) O_GTX_TX_N(3)	Connected to GPAC's PB_8MGT_*(7) signal.
GTX P0 Channel 0	I_GTX_RX_P(4) I_GTX_RX_N(4) O_GTX_TX_P(4) O_GTX_TX_N(4)	Connected to GPAC's P0_2MGT_*(1) signal.
GTX P0 channel 1	I_GTX_RX_P(5) I_GTX_RX_N(5) O_GTX_TX_P(5) O_GTX_TX_N(5)	Connected to GPAC's P0_2MGT_*(0) signal.
GTX BPM channel 0	I_GTX_RX_P(6) I_GTX_RX_N(6) O_GTX_TX_P(6) O_GTX_TX_N(6)	Connected to GPAC's BPM_4MGT_*(0) signal.
GTX BPM channel 1	I_GTX_RX_P(7) I_GTX_RX_N(7) O_GTX_TX_P(7) O_GTX_TX_N(7)	Connected to GPAC's BPM_4MGT_*(1) signal.
GTX BPM channel 2	I_GTX_RX_P(8) I_GTX_RX_N(8) O_GTX_TX_P(8) O_GTX_TX_N(8)	Connected to GPAC's BPM_4MGT_*(2) signal.
GTX BPM channel 3	I_GTX_RX_P(9) I_GTX_RX_N(9) O_GTX_TX_P(9) O_GTX_TX_N(9)	Connected to GPAC's BPM_4MGT_*(3) signal.

2.3 PLB Interface

2.3.1 Registers

The PLB register map is shown in Table 4 below. Each register is 32-bit wide. Addressing is bytewise. Parameters are read-write unless specified as read-only.

Table 4 – PLB register map

PLB Address offset	Bit range	Field name	Description
0x0	0	RST_QSFP0	Reset player 0 (including QSFP0 GTX link).
0x0	1	RST_QSFP1	Reset player 1 (including QSFP1 GTX link).
0x0	2	RST_QSFP2	Reset player 2 (including QSFP2 GTX link).
0x0	3	RST_QSFP3	Reset player 3 (including QSFP3 GTX link).
0x0	4	RST_P0.0	Reset player 4 (including P0.0 GTX link).
0x0	5	RST_P0.1	Reset player 5 (including P0.1 GTX link).
0x0	6	RST_BPM0	Reset player 6 (including BPM0 GTX link).
0x0	7	RST_BPM1	Reset player 7 (including BPM1 GTX link).
0x0	8	RST_BPM2	Reset player 8 (including BPM2 GTX link).
0x0	9	RST_BPM3	Reset player 9 (including BPM3 GTX link).
0x0	12	RST_LOS_CNT_QSFP0	Reset loss of sync counter for QSFP0 GTX link.
0x0	13	RST_LOS_CNT_QSFP1	Reset loss of sync counter for QSFP1 GTX link.
0x0	14	RST_LOS_CNT_QSFP2	Reset loss of sync counter for QSFP2 GTX link.
0x0	15	RST_LOS_CNT_QSFP3	Reset loss of sync counter for QSFP3 GTX link.
0x0	16	RST_LOS_CNT_P0.0	Reset loss of sync counter for P0.0 GTX link.
0x0	17	RST_LOS_CNT_P0.1	Reset loss of sync counter for P0.1 GTX link.
0x0	18	RST_LOS_CNT_BPM0	Reset loss of sync counter for BPM0 GTX link.
0x0	19	RST_LOS_CNT_BPM1	Reset loss of sync counter for BPM1 GTX link.
0x0	20	RST_LOS_CNT_BPM2	Reset loss of sync counter for BPM2 GTX link.
0x0	21	RST_LOS_CNT_BPM3	Reset loss of sync counter for BPM3 GTX link.
0x0	24	INT_TRIG_EN	Enable internal trigger. When '1', then each player is started independently whenever a 0->1 transition is detected on the PLAYER_START bit relative to that specific channel (see register at offset 0x14, bits 9:0). When '0', then all the player are started on the rising edge of the trigger coming from the timing component (external trigger delayed).
0x4	2:0	LOOPBACK_QSFP0	Loopback setting for QSFP0 GTX channel. See [3] chapter 9 for details.
0x4	6:4	LOOPBACK_QSFP1	Loopback setting for QSFP1 GTX channel. See [3] chapter 9 for details.
0x4	10:8	LOOPBACK_QSFP2	Loopback setting for QSFP2 GTX channel. See [3] chapter 9 for details.
0x4	14:12	LOOPBACK_QSFP3	Loopback setting for QSFP3 GTX channel. See [3] chapter 9 for details.
0x4	18:16	LOOPBACK_BPM0	Loopback setting for BPM0 GTX channel. See [3] chapter 9 for details.
0x4	22:20	LOOPBACK_BPM1	Loopback setting for BPM1 GTX channel. See [3] chapter 9 for details.
0x4	26:24	LOOPBACK_BPM2	Loopback setting for BPM2 GTX channel. See [3] chapter 9 for details.
0x4	30:28	LOOPBACK_BPM3	Loopback setting for BPM3 GTX channel. See [3] chapter 9 for details.

PLB Address offset	Bit range	Field name	Description
0x8	2:0	LOOPBACK_P0.0	Loopback setting for P0.0 GTX channel. See [3] chapter 9 for details.
0x8	6:4	LOOPBACK_P0.1	Loopback setting for P0.1 GTX channel. See [3] chapter 9 for details.
0x8	23:16	K_EOP	End-of-packet K-character used in IBFB protocol. Read-only.
0x8	31:24	K_SOP	Start-of-packet K-character used in IBFB protocol. Read-only.
0xC	0	LOCK_QSFP13	PLL lock for QSFP GTX tile (channels 1 and 3). See [3] for details. Read-only.
0xC	1	RST_DONE_QSFP3	Reset done for QSFP3 GTX link. See [3] for details. Read-only.
0xC	2	RST_DONE_QSFP1	Reset done for QSFP1 GTX link. See [3] for details. Read-only.
0xC	5:4	LOS_QSFP3	Status of Loss-of-sync FSM for QSFP3 GTX link. See [3], chapter 7 for details. Read-only.
0xC	7:6	LOS_QSFP1	Status of Loss-of-sync FSM for QSFP1 GTX link. See [3], chapter 7 for details. Read-only.
0xC	8	LOCK_QSFP02	PLL lock for QSFP GTX tile (channels 0 and 2). See [3] for details. Read-only.
0xC	9	RST_DONE_QSFP2	Reset done for QSFP2 GTX link. See [3] for details. Read-only.
0xC	10	RST_DONE_QSFP0	Reset done for QSFP0 GTX link. See [3] for details. Read-only.
0xC	13:12	LOS_QSFP2	Status of Loss-of-sync FSM for QSFP2 GTX link. See [3], chapter 7 for details. Read-only.
0xC	15:14	LOS_QSFP0	Status of Loss-of-sync FSM for QSFP0 GTX link. See [3], chapter 7 for details. Read-only.
0xC	16	LOCK_BPM01	PLL lock for BPM GTX tile (channels 0 and 1). See [3] for details. Read-only.
0xC	17	RST_DONE_BPM0	Reset done for BPM0 GTX link. See [3] for details. Read-only.
0xC	18	RST_DONE_BPM1	Reset done for BPM1 GTX link. See [3] for details. Read-only.
0xC	21:20	LOS_BPM0	Status of Loss-of-sync FSM for BPM0 GTX link. See [3], chapter 7 for details. Read-only.
0xC	23:22	LOS_BPM1	Status of Loss-of-sync FSM for BPM1 GTX link. See [3], chapter 7 for details. Read-only.
0xC	24	LOCK_BPM23	PLL lock for BPM GTX tile (channels 2 and 3). See [3] for details. Read-only.
0xC	25	RST_DONE_BPM2	Reset done for BPM2 GTX link. See [3] for details. Read-only.
0xC	26	RST_DONE_BPM3	Reset done for BPM3 GTX link. See [3] for details. Read-only.
0xC	29:28	LOS_BPM2	Status of Loss-of-sync FSM for BPM2 GTX link. See [3], chapter 7 for details. Read-only.
0xC	31:30	LOS_BPM3	Status of Loss-of-sync FSM for BPM3 GTX link. See [3], chapter 7 for details. Read-only.
0x10	0	LOCK_P0	PLL lock for P0 GTX tile (both channels). See [3] for details. Read-only.
0x10	9	RST_DONE_P0.0	Reset done for P0.0 GTX link. See [3] for details. Read-only.
0x10	8	RST_DONE_P0.1	Reset done for P0.1 GTX link. See [3] for details. Read-only.
0x10	5:4	LOS_P0.0	Status of Loss-of-sync FSM for P0.0 GTX link. See [3], chapter 7 for details. Read-only.

PLB Address offset	Bit range	Field name	Description
0x10	7:6	LOS_P0.1	Status of Loss-of-sync FSM for P0.1 GTX link. See [3], chapter 7 for details. Read-only.
0x10	8	PLAYER_EN_QSFP0	When 1, player 0 (QSFP0) is present in the core. Otherwise the player is not usable. Read-only.
0x10	9	PLAYER_EN_QSFP1	When 1, player 1 (QSFP1) is present in the core. Otherwise the player is not usable. Read-only.
0x10	10	PLAYER_EN_QSFP2	When 1, player 2 (QSFP2) is present in the core. Otherwise the player is not usable. Read-only.
0x10	11	PLAYER_EN_QSFP3	When 1, player 3 (QSFP3) is present in the core. Otherwise the player is not usable. Read-only.
0x10	12	PLAYER_EN_P00	When 1, player 4 (P0.0) is present in the core. Otherwise the player is not usable. Read-only.
0x10	13	PLAYER_EN_P01	When 1, player 5 (P0.1) is present in the core. Otherwise the player is not usable. Read-only.
0x10	14	PLAYER_EN_BPM0	When 1, player 6 (BPM0) is present in the core. Otherwise the player is not usable. Read-only.
0x10	15	PLAYER_EN_BPM1	When 1, player 7 (BPM1) is present in the core. Otherwise the player is not usable. Read-only.
0x10	16	PLAYER_EN_BPM2	When 1, player 8 (BPM2) is present in the core. Otherwise the player is not usable. Read-only.
0x10	17	PLAYER_EN_BPM3	When 1, player 9 (BPM3) is present in the core. Otherwise the player is not usable. Read-only.
0x10	20	USE_EXT_CLOCK	When 1, the core is using the external clock, otherwise the internal clock is being used. Read-only.
0x10	31:24	RAM_ADDR_W	Address width used for each player's RAM. The maximum number of packets that can be stored for each player is $(2^{RAM\_ADDR\_W}/4)-1$.
0x14	0	PLAYER_START_QSFP0	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 0 to start (QSFP0). Otherwise this bit has no effect.
0x14	1	PLAYER_START_QSFP1	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 1 to start (QSFP1). Otherwise this bit has no effect.
0x14	2	PLAYER_START_QSFP2	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 2 to start (QSFP2). Otherwise this bit has no effect.
0x14	3	PLAYER_START_QSFP3	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 3 to start (QSFP3). Otherwise this bit has no effect.
0x14	4	PLAYER_START_P00	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 4 to start (P0.0). Otherwise this bit has no effect.
0x14	5	PLAYER_START_P01	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 5 to start (P0.1). Otherwise this bit has no effect.
0x14	6	PLAYER_START_BPM0	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 6 to start (BPM0). Otherwise this bit has no effect.
0x14	7	PLAYER_START_BPM1	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 7 to start (BPM1). Otherwise this bit has no effect.
0x14	8	PLAYER_START_BPM2	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 8 to start (BPM2). Otherwise this bit has no effect.

PLB Address offset	Bit range	Field name	Description
0x14	9	PLAYER_START_BPM3	If the bit INT_TRIG_EN is set to 1, then a 0->1 transition on this field causes the player 9 to start (BPM3). Otherwise this bit has no effect.
0x14	10	RX_SYNC_QSFP0	Receiver synchronization status for QSFP0 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	11	RX_SYNC_QSFP1	Receiver synchronization status for QSFP1 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	12	RX_SYNC_QSFP2	Receiver synchronization status for QSFP2 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	13	RX_SYNC_QSFP3	Receiver synchronization status for QSFP3 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	14	RX_SYNC_P0.0	Receiver synchronization status for P0.0 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	15	RX_SYNC_P0.1	Receiver synchronization status for P0.1 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	16	RX_SYNC_BPM0	Receiver synchronization status for BPM0 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	17	RX_SYNC_BPM1	Receiver synchronization status for BPM1 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	18	RX_SYNC_BPM2	Receiver synchronization status for BPM2 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	19	RX_SYNC_BPM3	Receiver synchronization status for BPM3 GTX link. When '1' the RX channel is correctly synchronized. Read-only.
0x14	31:24	CTRL_EOS	END-OF-STREAM CTRL field: players stop if they read from the playback RAM a packet containing a CTRL field equal to this value.
0x18	31:0	N_SENT_PACKETS_QSFP0	Number of packets sent by player 0 (QSFP0) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x1C	31:0	N_SENT_PACKETS_QSFP1	Number of packets sent by player 1 (QSFP1) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x20	31:0	N_SENT_PACKETS_QSFP2	Number of packets sent by player 2 (QSFP2) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x24	31:0	N_SENT_PACKETS_QSFP3	Number of packets sent by player 3 (QSFP3) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x28	31:0	N_SENT_PACKETS_P0.0	Number of packets sent by player 4 (P0.0) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x2C	31:0	N_SENT_PACKETS_P0.1	Number of packets sent by player 5 (P0.1) since last trigger/start. This counter is reset whenever the player is started. Read-only.

PLB Address offset	Bit range	Field name	Description
0x30	31:0	N_SENT_PACKETS_BPM0	Number of packets sent by player 6 (BPM0) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x34	31:0	N_SENT_PACKETS_BPM1	Number of packets sent by player 7 (BPM1) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x38	31:0	N_SENT_PACKETS_BPM2	Number of packets sent by player 8 (BPM2) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x3C	31:0	N_SENT_PACKETS_BPM3	Number of packets sent by player 9 (BPM3) since last trigger/start. This counter is reset whenever the player is started. Read-only.
0x40	15:0	LOS_COUNTER_QSFP0	Loss-of-sync counter for QSFP0 GTX link. Can be reset with the RST_LOS_CNT_QSFP0 command. Read-only.
0x40	31:16	LOS_COUNTER_QSFP1	Loss-of-sync counter for QSFP1 GTX link. Can be reset with the RST_LOS_CNT_QSFP1 command. Read-only.
0x44	15:0	LOS_COUNTER_QSFP2	Loss-of-sync counter for QSFP2 GTX link. Can be reset with the RST_LOS_CNT_QSFP2 command. Read-only.
0x44	31:16	LOS_COUNTER_QSFP3	Loss-of-sync counter for QSFP3 GTX link. Can be reset with the RST_LOS_CNT_QSFP3 command. Read-only.
0x48	15:0	LOS_COUNTER_P0.0	Loss-of-sync counter for P0.0 GTX link. Can be reset with the RST_LOS_CNT_P0.0 command. Read-only.
0x48	31:16	LOS_COUNTER_P0.1	Loss-of-sync counter for P0.1 GTX link. Can be reset with the RST_LOS_CNT_P0.1 command. Read-only.
0x4C	15:0	LOS_COUNTER_BPM0	Loss-of-sync counter for BPM0 GTX link. Can be reset with the RST_LOS_CNT_BPM0 command. Read-only.
0x4C	31:16	LOS_COUNTER_BPM1	Loss-of-sync counter for BPM1 GTX link. Can be reset with the RST_LOS_CNT_BPM1 command. Read-only.
0x50	15:0	LOS_COUNTER_BPM2	Loss-of-sync counter for BPM2 GTX link. Can be reset with the RST_LOS_CNT_BPM2 command. Read-only.
0x50	31:16	LOS_COUNTER_BPM3	Loss-of-sync counter for BPM3 GTX link. Can be reset with the RST_LOS_CNT_BPM3 command. Read-only.
0x54	0	GLOBAL_TRG_ENA	Timing component: *global_trg_ena* input (cfr. [???] for details).
0x54	8	TRG_MODE	Timing component: *trg_mode* input (cfr. [???] for details).
0x54	18:16	TRG_SOURCE	Timing component: *trg_source* input (cfr. [???] for details).
0x58	27:0	B_DELAY	Timing component: *b_delay* input (cfr. [???] for details).
0x5C	15:0	B_NUMBER	Timing component: *b_number* input (cfr. [???] for details).
0x5C	31:16	B_SPACE	Timing component: *b_space* input (cfr. [???] for details).
0x60	2:0	TRG_RATE	Timing component: *trg_rate* input (cfr. [???] for details).

PLB Address offset	Bit range	Field name	Description
0x64	ANY	TRG_ONCE	Timing component: any write to this register generates a pulse on the timing component's *trg_once* input (cfr. [???] for details).
0x68	0	EXT_TRG_MISSING	Timing component: *ext_trg_missing* output (cfr. [???] for details).
0x68	8	READ_READY	Timing component: *read_ready* output (cfr. [???] for details).
0x7C	31_0	FW_VERSION	Component's version number (unsigned integer, incremental). Current value 0x3.

2.3.2 RAMs

The PLB slave provides also a RAM address space used to load the data to be played into the players' RAMs. All the players share a common address space. The address' top 4 bits are used to select one of the players. The lower C_PLAYER_RAM_ADDR_W bits are used to address the content of each player. Each player's address space is divided in 4 sections: the first section contains the timestamps for each packet. The second section contains the CTRL, BPM and BUCKET fields. The third and fourth section hold respectively the X-position and Y-position data fields.

The addressing scheme is detailed in Table 5. For details on the IBFB packet format, refer to §2.4.1.

Be aware that the data is stored inside the RAM in another order, as described in §2.4.1. This particular addressing scheme is obtained by reorganization of the address bits on the RAM's write port.

Table 5 - Player's RAM addressing scheme.

FULL ADDRESS (PLAYER_RAM_ADDR_W+4)		
PLAYER(4)	**SECTION(2)**	**PACKET(PLAYER_RAM_ADDR_W-2)**
Select player (0x0 to 0x9). The section is accessible only if the player is instantiated (proper C_PLAYER_EN bit shall be set)	Select which data field to access (each field is 32-bit wide): **00:** timestamp. Measured in clock cycles from the internal trigger (shall be monotonically increasing) **01:** dword containing the fields CTRL(8), BPM(8) and BUCKET(16) **10:** X-position **11:** Y-position	Sequential packet identifier.

2.4 Functional description

The core's main function is implemented in the *ibfb_packet_player* component. This component is replicated several times, according to the number of channels needed. The core's top level just connects the *ibfb_packet_player* instances to the proper GTX channels and provides the external PLB interface.

2.4.1 IBFB_packet_player (single channel)

The **ibfb_packet_player** component has the basic task of reading data packets from a RAM and outputting them at a specific time with respect to a start trigger. The output delay is read from the RAM along with the data. The component's architecture is shown in Figure 4.

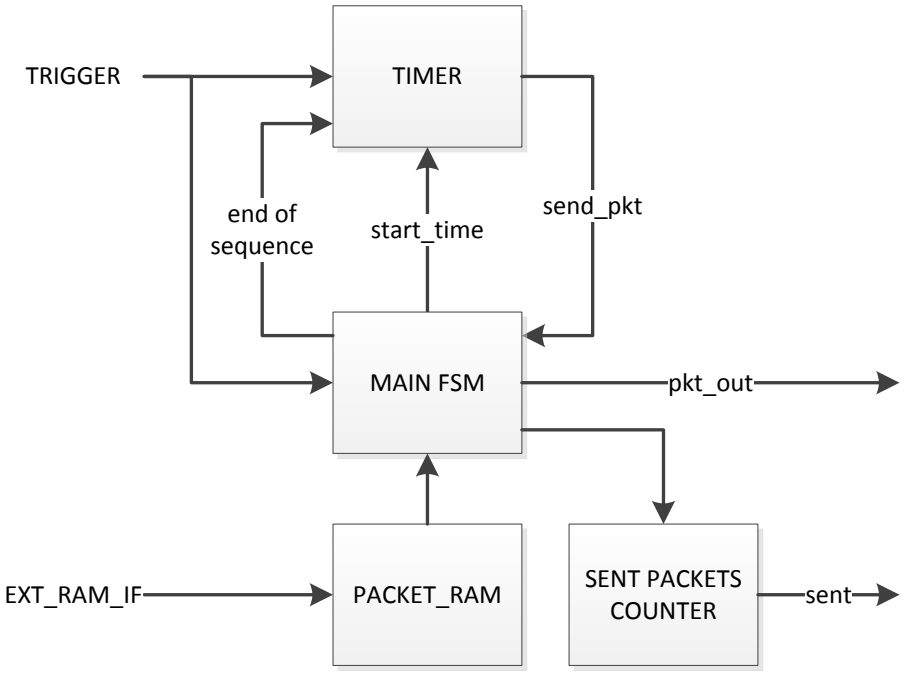

Figure 4 – IBFB packet player's architecture

A *Timer* is started on the rising edge of the input *Trigger*. It runs continuously until the main FSM signals the end of a packet sequence (*end_of_sequence*). The *Timer* is used to generate a *send_packet* signal. The *send_packet* is asserted whenever the timer is greater than the value *start_time* provided by the *MAIN_FSM*.
The *MAIN_FSM* is in charge of reading data from the *PACKET_RAM* and outputting it with the right timing. Figure 5 shows the FSM's state diagram.

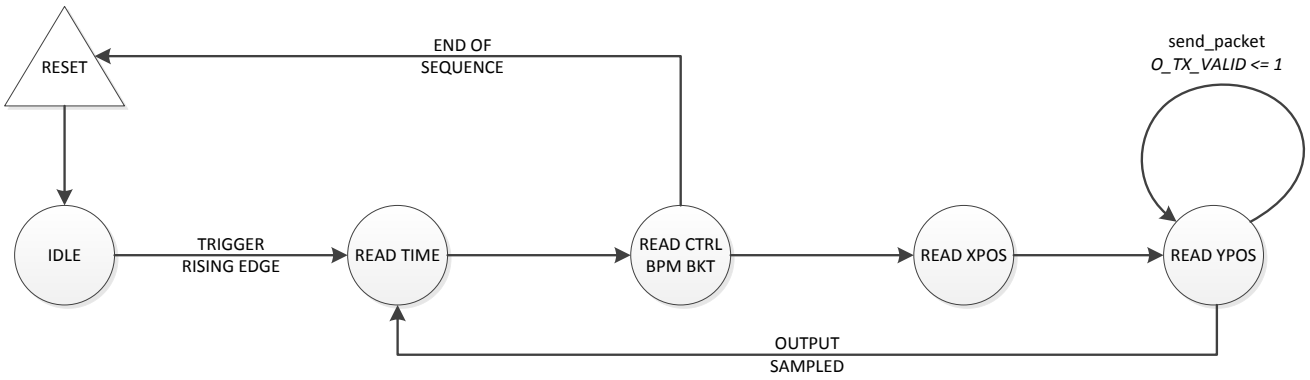

Figure 5 - State diagram for ibfb_packet_player MAIN_FSM.

After reset the FSM waits for the trigger's rising edge, then it starts to read packets from the RAM's address 0x0.

Packets are stored in the RAM sequentially, as 4 consecutive 32-bit words. The 1st word contains the packet's transmit delay with respect to the trigger, the 2nd word contains the CTRL, BPM and BUCKET fields, while the 3rd and 4th words contain the X-position and Y-position respectively. The packet structure in RAM is summarized in Table 6.

Table 6 - IBFB packet as stored in player's RAM

offset	Content			
	31:24	23:16	15:8	7:0
0x0	Timestamp			
0x4	CTRL	BPM	BUCKET	
0x8	X POSITION			
0xC	Y POSITION			

Timestamp is read first and stored in the *start_time* register. This value is compared to the current value of the timer and the signal *send_packet* is asserted as soon as the timer is found to be greater than it.
Then the CTRL field is read. If the CTRL field is found to be equal to the END_OF_SEQUENCE byte, then the FSM is reset and the player stops, waiting for the next trigger. Otherwise the fields CTRL, BPM and BUCKET are stored followed by the X and Y position values.
When all the packet content has been read, the FSM waits until the signal *send_packet* is asserted. Then data is made available on the output port and the signal *o_tx_valid* is asserted. The FSM then waits until the output data is sampled (*i_sop = 1*).
After that, the process repeats by reading the next timestamp.

Figure 6 shows the timing diagram of the playback process for a case in which the RAM contains a single packet.

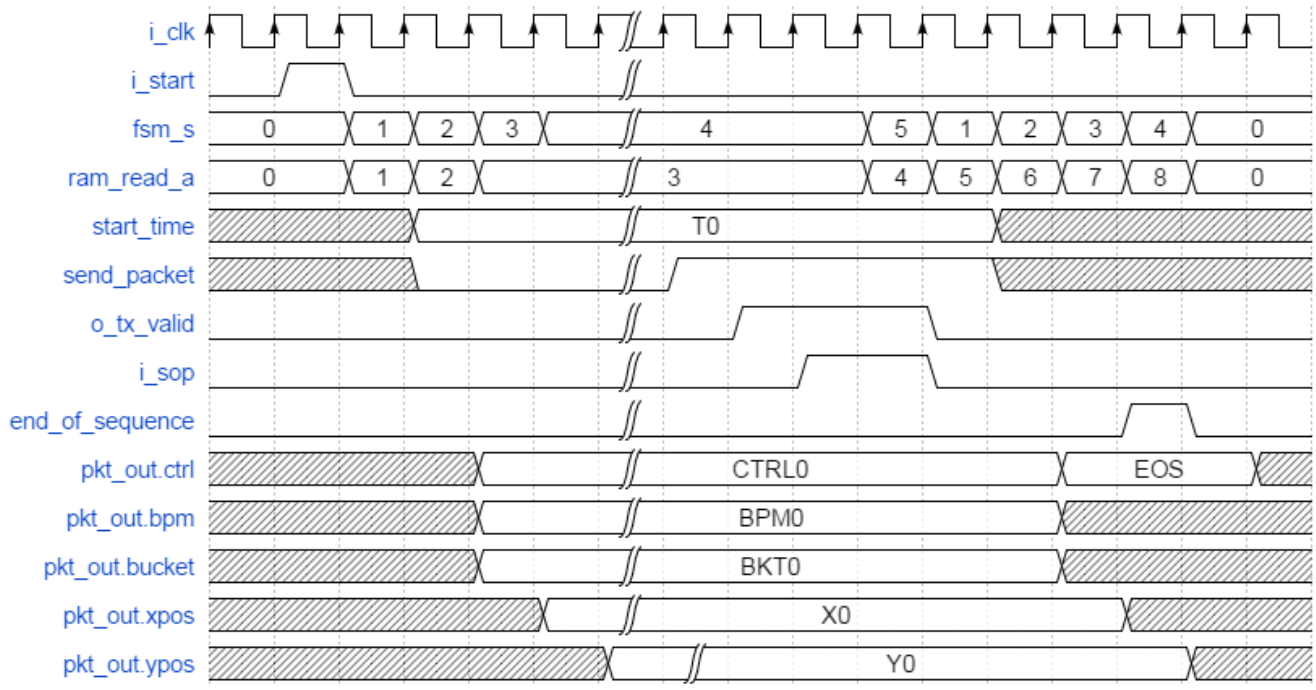

Figure 6- Timing diagram of packet playback (RAM containing a single packet)

2.5 FPGA Resources

Table 7 – Resource utilization (5 players)

Version	Flip Flops	LUTs	LUTRAMs	BRAMs	GTX
Virtex-5	3653	2847	33	90	5

2.6 Design constraints

The core has no specific constraints.