

Weighted Fair Queue (WFQ)

Author: David Gabbay
DoGav Systems Ltd.
January, 2006

Contents

1. References.....	4
2. Abbreviations.....	4
3. Introduction.....	5
4. Host – CPM Interface Definitions	6
4.1. FCC Parameter Ram	7
4.2. WFQ Structure.....	8
4.3. Queue Structure	9
4.4. Clear Deficit Command.....	10
5. Installation.....	11
5.1. Package contents.....	11
5.2. Trap usage.....	11

Revision History

<i>Version</i>	<i>Date</i>	<i>Modified by</i>	<i>Status</i>	<i>Approver</i>
1.0	17-Jan-06	Oded Gabbay	Beta	David Gabbay
1.1	24-Feb-06	Amy Dyson	Final	David Gabbay

List of Tables

Table 1. Abbreviations used in this document.....	4
Table 2. FCC Parameter RAM.....	7
Table 3. WFQ Structure.....	8
Table 4. WFQ_CTRL_Qx Fields.....	9
Table 5. Queue Structure	9
Table 6. Clear Deficit Commands Description.....	10

List of Figures

Figure 1. WFQ RAM organization.....	6
Figure 2. WFQ_CTRL_Qx Definition.....	8
Figure 3. Clear Deficit Command Format	10

1. References

1. MPC8280 Reference Manual Rev 0 03/2004

2. Abbreviations

Table 1. Abbreviations used in this document.

Abbr.	Description
BD	Buffer Descriptor
CRC	Cyclic Redundancy Check
LUT	Look-Up Table
Rx	Receive
RxBD	Receive Buffer Descriptor
Tx	Transmit
TxBD	Transmit Buffer Descriptor
uCode	PowerQUICC's CPM Microcode
WFQ	Weighted Fair Queue

3. Introduction

This uCode package applies to the FCC's Ethernet Protocol. It manages transmit queues by implementing the well-known Weighted Fair Queue (WFQ), supporting up to eight transmit queues. When the FCC is ready to transmit a new frame, the uCode uses the WFQ algorithm to choose a queue from which to pull a frame.

WFQ is a flow-based queuing algorithm that schedules low-volume traffic first. High-volume traffic shares the remaining bandwidth. Each flow is assigned a weight, where lower weights have higher priority and are serviced first.

WFQ ensures that queues are not starved for bandwidth and that traffic gets predictable service. Low-volume traffic streams comprise the majority of traffic and receive preferential service. High-volume traffic streams share the remaining capacity proportionally.

WFQ never wastes bandwidth. If there's no higher priority traffic, lower priority traffic is forwarded.

WFQ Package Main Features:

- Supports up to eight queues.
- Weight (cost) of queue can be modified dynamically.
- "Clear all queues' deficits" or "Clear specific queue's deficit" commands.
- Statistics: 32-bit counter per queue counts the number of frames transmitted.

4. Host – CPM Interface Definitions

The WFQ mechanism defines a structure and an array of structures that must be located in the DPRAM. Figure 1 illustrates the WFQ RAM organization.

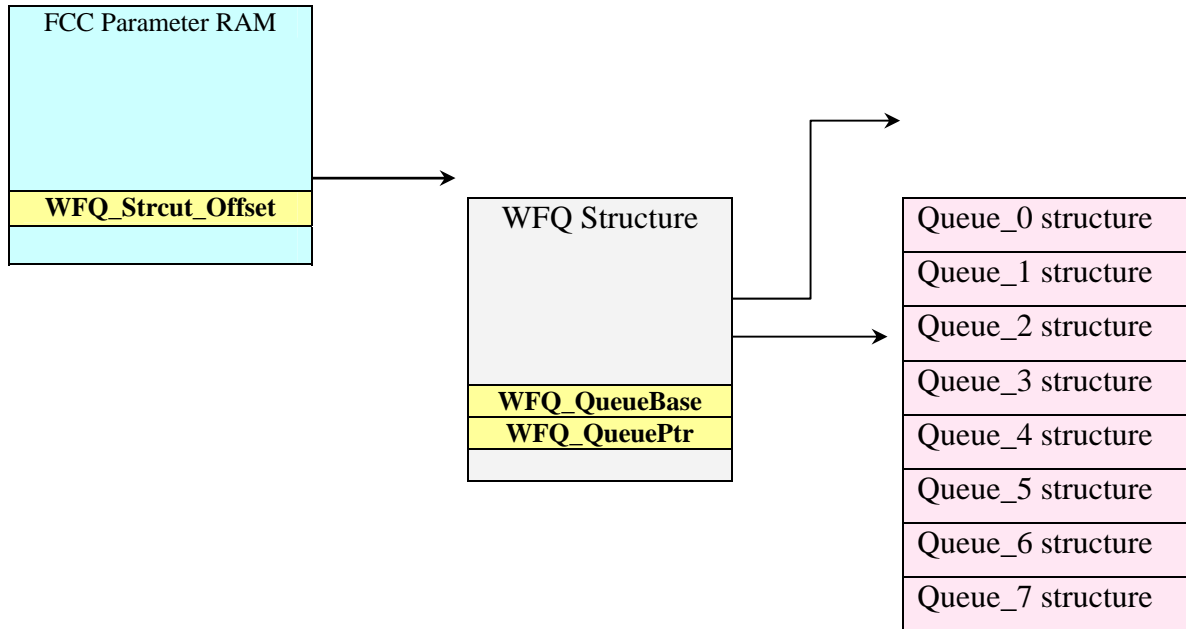


Figure 1. WFQ RAM organization

4.1. FCC Parameter Ram

To support WFQ, a new parameter has been added to the FCC Ethernet parameter RAM. This two-byte parameter, located at offset 0xFC-0xFD, serves as a pointer to the WFQ structure in the DPRAM.

Table 2 describes the additional parameter required by the uCode.

Table 2. FCC Parameter RAM

Offset	Name	Width	Description
0x00 – 0xFB	-	-	Same as in mpc8280 manual (Ref. 1).
0xFC	WFQ_Strcut_Offset	Hword	Offset from the DPRAM base address to the WFQ structure. It <u>must</u> be divisible by 128 .
0xFE- 0xFF	-	-	Same as in mpc8280 manual (Ref. 1).

4.2. WFQ Structure

The WFQ structure must be located in the DPRAM at an offset divisible by 128.

Table 3 describes the WFQ structure.

Table 3. WFQ Structure

Offset	Name	Width	Description
0x00	WFQ_TX_Q0	Word	Statistics: Number of transmitted frames Q0. Initialize to zero.
0x04	WFQ_TX_Q1	Word	Statistics: Number of transmitted frames Q1. Initialize to zero.
0x08	WFQ_TX_Q2	Word	Statistics: Number of transmitted frames Q2. Initialize to zero.
0x0C	WFQ_TX_Q3	Word	Statistics: Number of transmitted frames Q3. Initialize to zero.
0x10	WFQ_TX_Q4	Word	Statistics: Number of transmitted frames Q4. Initialize to zero.
0x14	WFQ_TX_Q5	Word	Statistics: Number of transmitted frames Q5. Initialize to zero.
0x18	WFQ_TX_Q6	Word	Statistics: Number of transmitted frames Q6. Initialize to zero.
0x1C	WFQ_TX_Q7	Word	Statistics: Number of transmitted frames Q7. Initialize to zero.
0x20	WFQ_CTRL_Q0	Hword	Control half-word for queue 0.
0x22	WFQ_CTRL_Q1	Hword	Control half-word for queue 1.
0x24	WFQ_CTRL_Q2	Hword	Control half-word for queue 2.
0x26	WFQ_CTRL_Q3	Hword	Control half-word for queue 3.
0x28	WFQ_CTRL_Q4	Hword	Control half-word for queue 4.
0x2A	WFQ_CTRL_Q5	Hword	Control half-word for queue 5.
0x2C	WFQ_CTRL_Q6	Hword	Control half-word for queue 6.
0x2E	WFQ_CTRL_Q7	Hword	Control half-word for queue 7.
0x30	WFQ_SortSpaceOpen	8 bytes	Microcode managed parameter; initialize to zero.
0x38	WFQ_SortSpaceBlocked	8 bytes	Microcode managed parameter; initialize to zero.
0x40	WFQ_Sort_Flag	Word	Microcode managed parameter; initialize to zero.
0x44	WFQ_Save_LoopReg	Word	Microcode managed parameter; initialize to zero.
0x48	WFQ_QueueBase	Hword	Base address of the queue's array. Offset from the start of the DPRAM base address. Must be divisible by 16.
0x4A	WFQ_QueuePtr	Hword	Pointer to current queue structure. Initialize to <i>WFQ_QueueBase</i> value.
0x4C-0x7F	Reserved	-	Reserved. Initialize to zero.

Figure 2 shows the layout of **WFQ_CTRL_Qx**. Table 4 further defines its fields.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reserved			last	Reserved											rdy

Figure 2. WFQ_CTRL_Qx Definition

Where:

Table 4. WFQ_CTRL_Qx Fields

Bit	Name	Description
0-2	Reserved	Reserved. Initialize to zero.
3	last	Last entry. For example, if the application sets up 6 queues, this bit should be set in WFQ_CTRL_Q5.
4-14	Reserved	Reserved. Initialize to zero.
15	rdy	Current BD has a ready bit set. Every time the application sets the ready bit in the TxBD, it must also set this bit. This bit is cleared by the CPM when no ready TxBD is encountered. This bit minimizes bus transactions.

4.3. Queue Structure

In addition to the WFQ structure, an array of eight queue structures is required. The array must be located in the DRAM at an offset divisible by 16. Table 5 shows the queue structure.

Table 5. Queue Structure

Offset	Name	Width	Description
0x00	Q_TBASE	Word	BD ring base address (must be aligned to 8).
0x04	Q_TBDPTR	Word	Current BD address. Initialize to <i>Q_TBASE</i> .
0x08	Q_AccumWeight	Word	Current BD weight (deficit). Initialize to zero.
0x0C	Q_Weight	Hword	Queue weight (cost per 32 Tx bytes).
0x0E	Q_Num	Hword	Queue number (value 0...7).

4.4. Clear Deficit Command

Two commands may be used to clear queue deficits:

- One command clears all queues' deficits.
- One command clears a specific queue's deficit.

The command should be issued to the CPM's CPCR (offset 0x1_01a8). Figure 3 shows the command format.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LST	Page				Sub-block (SBC)				MBZ			A/Q	FLG	MBZ						Queue			MBZ		0xE						
R/W																															
0x119C0																															

Figure 3. Clear Deficit Command Format

Table 6 further defines the command format.

Table 6. Clear Deficit Commands Description

Bit	Name	Description
0	RST	Reset. See Ref 1.
1-5	Page	0x04 for FCC1, 0x05 for FCC2, 0x06 for FCC3.
6-10	SBC	0x10 for FCC1, 0x11 for FCC2, 0x12 for FCC3.
11-13	MBZ	Must be zero.
14	A/Q	All / Specific Queue 0 = Clear all queues' deficits 1 = Clear a specific queue deficit (specify queue No. in field queue)
15	FLG	Flag. See Ref 1.
16-22	MBZ	Must be zero.
23-25	Queue	Specifies the queue number when a 'clear specific queue deficit' command is issued. If a 'clear all queues deficit' is issued, keep this field cleared.
26-27	MBZ	Must be zero.
28-31	CMD	Command. Must be 0xE ¹ .

¹ This command value was reserved when the uCode this module was defined. Freescale™ may assign this value for another purpose. In that case, DoGav Systems will use a different value.

For example:

To clear all queues' deficits for FCC2 issue: 0x1621_000e.

To clear queue 4's deficit issue: 0x1623_014e.

5. Installation

5.1. Package contents

This package comes as a ZIP compressed file containing the following files:

1. *InstallUcode.c* – This file contains two C functions, one for installing the uCode and the other for uninstalling it. Call *InstallUcode* to install the uCode. Call *UninstallUcode* to uninstall the uCode. For more details, refer to the description in *InstallUcode.h*
2. *InstallUcode.h* – This file contains function prototypes.
3. *PQ2_WFQ_uCode.h* – This file contains the C array representing the uCode and defines the trap addresses.
4. *BasicDef.h* – This file contains definitions and typedefs required to compile the installation function.
5. *WFQ Microcode User Guide.pdf* – This document.

5.2. Trap usage

This uCode package uses 4 traps.