To: **T-10/SPI-4 Working Group**
From: **Gary Dickinson, Greg Goodemote, Chuck Micalizzi, and Richard Moore**
QLogic Corp.
Date: **October 26, 1999**

FAST-160 Buffer Credit Transfer Mechanism

This proposal describes a flow control scheme for Fast-160 SCSI using buffer credits. The traditional synchronous REQ or ACK signal is susceptible to first-pulse distortion. Replacing this signal with a lower-frequency signal can avoid this problem.

The description here is based on the assumption of a free-running clock with a data transfer enable. However, the buffer credit mechanism is compatible with a more traditional SCSI transfer mechanism that uses unqualified clocking.

Advantages:

The buffer credit scheme has the following advantages:

1. The buffer credit signal (ACK or REQ) is run-length limited, which reduces the amount of ISI. In addition, this signal runs at a much lower frequency than the data transfer, so that even if ISI is present, the low frequency eliminates adverse effects.
2. The buffer credit signal is filtered prior to edge detection. A 10 ns or even longer filter at the receiver is acceptable. This provides noise immunity on this signal.
3. Extensive pre-buffering of incoming data is not required.
4. The buffer credit mechanism is combined with 4-byte data transfer granularity, so that there are no large gaps in the data transfers.
5. Mapping packetized protocol to buffer credits is simplified by not allowing buffer credits to extend over packet boundaries.

Rules for Buffer Credits:

Buffer credits are transmitted in the opposite direction of the data flow, from the SCSI device receiving data to the SCSI device sending data, to indicate the readiness of the SCSI device receiving data.

The enabling of buffer credits and the amount of data represented by each buffer credit are negotiated through the PPR message. Since the concept of REQ/ACK offset is not applicable to buffer credit transfers, it is possible to replace the offset field of the PPR message with a buffer credit size field when buffer credit transfers are enabled.

A device's input data buffers must be capable of receiving twice the negotiated buffer credit size without pauses at the negotiated transfer rate.

At Fast-160 rates, a credit size of 128 bytes corresponds to 400 ns, which is greater than a round-trip bus delay. Thus 128 is the recommended minimum credit size for Fast-160.

A buffer credit is issued by a device when it is capable of receiving an amount of data equal to the negotiated buffer credit size. A device shall not have issued more than two outstanding buffer credits at any given time. It shall issue the first buffer credit after the beginning of DTDI or DTDO phase when it is ready to receive data. The device shall issue each subsequent buffer credit after the transfer of data for the most recent buffer credit has started and the device is capable of accepting the remainder of the first buffer credit plus the entirety of the second. A buffer credit is retired when the negotiated buffer credit size has been transferred, or when a packet boundary is reached if information unit transfers are enabled, or when a CRC is transmitted if information unit transfers are disabled.

An initiator shall not issue a buffer credit following the last buffer credit of an information unit in DTDI phase until all data transfers for that information unit have been completed and the CRC has been verified. In the event the initiator creates an attention condition, this restriction facilitates the identification of the information unit to which the attention condition applies.

The device sending buffer credits must assert the buffer credit signal for a period of at least 50 ns before negating it. The recommended assertion period for the buffer credit signal is equal to the time that would be required to transfer one half of a buffer credit at the negotiated rate (this recommendation gives a 50% duty cycle if the transfer is continuous).

The device sending data (and receiving buffer credits) shall filter the buffer credit signal for at least 10 ns, but not more than 25 ns, before recognizing a new buffer credit.

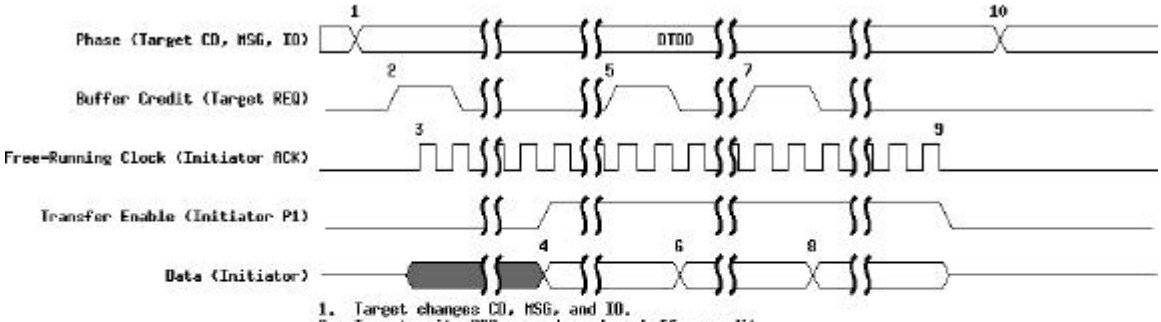
For each buffer credit received, the device sending data sends an amount of data exactly equal to the buffer credit size, unless (a) information units are enabled, and the end of a packet is reached, or (b) information units are disabled, and a CRC is transmitted.

CRC bytes are not counted against the buffer credit (because the receiving device does not require any buffer space for the CRC; this also allows buffer credit sizes equal to powers of two to map readily to typical packet or CRC block sizes).

An initiator may create an attention condition for a DTDI transfer by asserting ATN ??? ns prior to the next buffer credit (ACK) assertion. When the target detects ACK and ATN simultaneously true, it shall honor the attention condition.

DTDO Phase:

In DTDO phase, REQ acts as a buffer credit signal, ACK acts as the clock, and P(1) acts as the data enable.



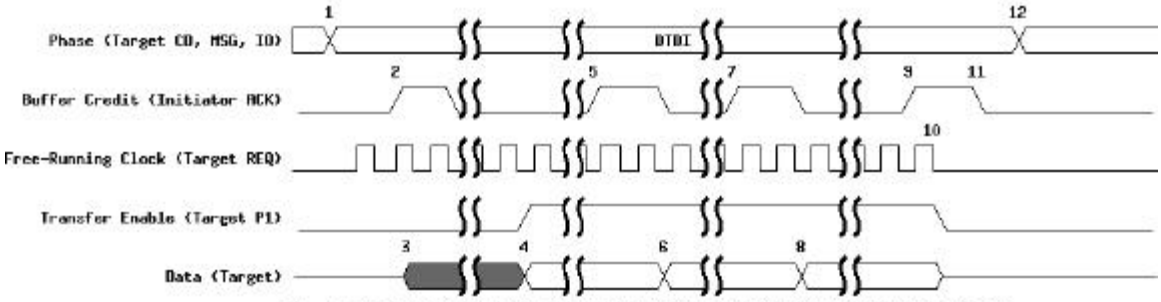
1. Target changes CD, MSG, and IO.
2. Target waits ??? ns and sends a buffer credit.
3. Initiator detects CD, MSG, and IO valid for DTDO and detects the buffer credit, then begins transmitting free-running clock and data training pattern.
4. Initiator enables transfer and starts first buffer credit transfer.
5. Target detects transfer enable and sends another buffer credit.
6. Initiator detects second buffer credit and starts second buffer credit transfer.
7. Target detects start of second buffer credit transfer and sends a third buffer credit.
8. Initiator detects third buffer credit and starts third buffer credit transfer.
9. Initiator runs out of buffer credits and stops clock.

?NOTE: If the target were to issue another buffer credit after the initiator ?stops its free-running clock, then the initiator must restart its clock for ???? cycles before resuming data transfer.

10. Target detects clock stopped and changes phase.

DTDI Phase:

In DTDI phase, ACK acts as a buffer credit signal, REQ acts as the clock, and P(1) acts as the data enable.



1. Target changes CD, MSG, and IO; after ??? ns target begins sending idle REQs.
2. Initiator detects phase change and sends a buffer credit.
3. Target detects buffer credit and begins transmitting a data training pattern.
4. Target enables transfer and starts first buffer credit transfer.
5. Initiator detects transfer enable and sends another buffer credit.
6. Target detects second buffer credit and starts second buffer credit transfer.
7. Initiator detects start of second buffer credit transfer and sends a third buffer credit.
8. Target detects third buffer credit and starts third buffer credit transfer.
9. Initiator detects start of third buffer credit transfer and sends a fourth buffer credit.
10. Target has no more data to send and stops clock.
11. Initiator negates its buffer credit signal (if asserted) within ??? ns of detecting clock stopped.
12. Target waits ??? ns after stopping clock and changes CD, MSG, and IO (last buffer credit is ignored).