To: **Improved SCSI Protocol Ad-Hoc Group**
From: **Andrew Wilson (DWilson@corp.adaptec.com)**
Subject: **Performance Estimates for Low Fat Protocol**
Date: **Monday, July 14, 1997**

1 Introduction

As the synchronous data transfer portion of a SCSI request becomes faster and faster, the time spent in SCSI protocol overhead becomes an ever increasing fraction of the total time. While substantial reductions in overhead can still be obtained by improving SCSI device implementations, there is a lower limit built into the SCSI protocol which will restrict performance at Fast-80 speeds. In response to this, Adaptec is proposing a set of protocol enhancements, collectively referred to as the Low Fat Protocol (LFP) which seek to further reduce the minimum SCSI protocol overhead, thus allowing efficient utilization of the Fast-80 SCSI bus. This white paper will demonstrate the benefits that can be obtained from these protocol enhancements and show how their inclusion results in reasonable SCSI efficiency, even for smaller transfer lengths.

2 Modeling Approach and Assumptions

Adherence to the SCSI-2 or SCSI-3 interlocked protocol specifications results in a variety of protocol mandated time periods which impose a lower bound on the amount of overhead incurred by a transaction. The protocol also requires frequent handshakes between Initiator and Target that incur transmission latencies and result in additional overhead. All this overhead not only delays completion of commands, but it represents wasted time on the bus, thus reducing the realizable bus bandwidth. Developing Initiator and Target implementations that closely approach these minimums are desirable for good performance.

Figure 1 shows calculations for the minimum overhead allowed by an ideal implementation of the SCSI protocol. As is done throughout this paper, a cable length of six meters is assumed.

Figure 1 divides the SCSI transaction into four sections: a startup section where commands and messages are passed to the Target, a disconnect / reconnect section incurred while the target seeks to the correct sector (Seek Disconnect), a disconnect / reconnect section during data transfer (Data Disconnect) and finally a data transfer and completion section. The data disconnect adds more overhead than a seek disconnect, because of the necessity of restarting the data transfer and saving pointers. Using these four sections, minimum overhead values for read transactions with any number of disconnects can be quickly calculated.

The assumed commands are long reads, with tagged queuing. Thus the COMMAND phase transfers ten bytes to the Target, and the MESSAGE OUT phase transfers three bytes. Similarly, after RESELECTION, the Target will have to indicate which transaction to resume with a one byte IDENTIFY and a two byte QUEUE TAG message, for a total of three bytes of MESSAGE IN.

	Overheads (in Nanoseconds)		
Phase	**Target**	**Initiator**	**Sys Tot**
BUS FREE			1,200
ARBITRATION		3,600	3,600
SELECTION	430	90	520
MSG OUT (Ident & tag)	580	234	814
COMMAND	1,000	780	1,780
Out to In Transition	400		400
SubTotal: Startup	**2,410**	**4,704**	**8,314**
MSG IN (Disconnect)	478	60	538
BUS FREE			1,200
ARBITRATION	3,600		3,600
RESELECTION	90	430	520
MSG IN (Ident & Tag)	634	180	814
SubTotal: Seek Disconnect	**4,802**	**670**	**6,672**
Startof Data Phase	400		400
MSG IN (Disc. & save ptrs)	556	120	676
BUS FREE			1,200
ARBITRATION	3,600		3,600
RESELECTION	90	430	520
MSG IN (Ident & Tag)	634	180	814
SubTotal: Data Disconnect	**5,280**	**730**	**7,210**
Startof Data Phase	400		400
STATUS	478	60	538
MSG IN (cmd complete)	478	60	538
SubTotal: Completion	**1,356**	**120**	**1,476**
Total Minimum Overhead	**13,848**	**6,224**	**23,672**

Figure 1: SCSI-2 Specification Minimum Overheads

Figure 1 shows the overhead produced by the SCSI protocol for a typical transaction which includes a pair of disconnects and operates over a six meter cable. This could be thought of as a theoretically ideal implementation with infinitely fast transceivers and internal sequencing logic. As logic circuitry gets faster, real implementations will approach this ideal, though they won't reach it. Since the transmission speed in a wire is fixed by physical laws, a propagation delay appropriate for the assumed six meter cable has been added for each instance where one device is waiting for a signal from the other. For example, to send a message or command byte using asynchronous protocol involves four such waits, resulting in 4 * 6 meters worth of delay, on top of a SCSI mandated Data Hold Time. The amount of time required to asynchronously transmit one byte is the sum of 3 nanoseconds wire skew delay, 15 nanoseconds system deskew delay, and 4 * 30 nanoseconds of propagation delay, for a total of 138 nanoseconds.

3 Protocol Enhancements

As mentioned in the introduction, the practical minimum overhead, though small compared to current implementations, is still large compared to the time it takes to transfer a few kilobytes of data at Fast-80 SCSI speeds. Thus, Adaptec is proposing three protocol enhancements to reduce the minimum practical overhead further. The effects of the three protocol enhancements are shown in Figure 2 for the same two disconnect transfer used in Figure 1. Figure 2 assumes the SCSI-2 specification mandated minimum overhead of Figure 1, and shows the improved overhead for each of the three proposals individually and taken all together.

Protocol Enhancements:	SCSI-2	LFP Proposals (ave. over 8 read cmds, in Nanoseconds)					
	Spec.	SMS	BCP	QAS	Total Effect		
	Sys Tot	Sys Tot	Sys Tot	Sys Tot	Target	Initiator	Sys Tot
Phase							
BUS FREE	1,200	1,200	150	0			0
ARBITRATION	3,600	3,600	450	1,138	89	54	142
SELECTION	520	520	65	690	54	33	86
MSG OUT (Ident & tag)	814	814	714	814	84	630	714
COMMAND	1,780	1,780		1,780			
Out to In Transition	400	400	50	400	50	0	50
SubTotal: Startup	**8,314**	**8,314**	**1,429**	**4,822**	**276**	**716**	**992**
MSG IN (Disconnect)	538	538	67	538	60	8	67
BUS FREE	1,200	1,200	1,200	0			0
ARBITRATION	3,600	3,600	3,600	1,138	1,138		1,138
RESELECTION	520	520	520	720	490	230	720
MSG IN (Ident & Tag)	814	814	814	814	634	180	814
SubTotal: Seek Disconnect	**6,672**	**6,672**	**6,201**	**3,210**	**2,322**	**418**	**2,739**
Startof Data Phase	400	400	400	400	400		400
MSG IN (Disc. & save ptrs)	676	538	676	676	478	60	538
BUS FREE	1,200	1,200	1,200	0			0
ARBITRATION	3,600	3,600	3,600	1,138	1,138		1,138
RESELECTION	520	520	520	720	490	230	720
MSG IN (Ident & Tag)	814	814	814	814	634	180	814
SubTotal: Data Disconnect	**7,210**	**7,072**	**7,210**	**3,748**	**3,140**	**470**	**3,610**
Startof Data Phase	400	400	400	400	400		400
STATUS	538	538	538	538	478	60	538
MSG IN (cmd complete)	538	0	538	538	0	0	0
SubTotal: Completion	**1,476**	**938**	**1,476**	**1,476**	**878**	**60**	**938**
Total Minimum Overhead	**23,672**	**22,996**	**16,316**	**13,256**	**6,616**	**1,664**	**8,280**
Savings		**676**	**7,356**	**10,416**			**15,393**

Figure 2: Details of Advanced Protocol Benefits (shaded entries indicate differences)

The Status / Message Simplification (SMS) proposal (the second column of Figure 2) consists of two parts, one which creates a new message SAVE DATA POINTERS AND DISCONNECT, eliminating the need to send separate DISCONNECT and SAVE DATA POINTERS messages and one which eliminates the COMMAND COMPLETE message on good status. The SAVE DATA POINTERS AND DISCONNECT message results in a reduction of 278 nanoseconds for each Data Disconnect operation, according to our model. The elimination of the final COMMAND COMPLETE message results in a savings of 678 nanoseconds per transaction, because both a message byte transfer and a phase change delay are avoided.

The third column of Figure 2 shows the benefits of the Broadcast Command Packet (BCP) proposal. This proposal mostly affects the startup portion, by expediting the COMMAND and MESSAGE OUT phases and by batching several commands together under one bus arbitration. The modeled overhead assumes that we can batch 8 commands into one BCP phase, so the average arbitration and selection times are reduced to 1/8 of their normal values. The one other savings is a reduction in DISCONNECT messages occurring as part of the Seek Disconnect portion of the transfer, because all but the last disconnect is implied by the protocol.

I have chosen to account for the time spent transmitting the BCP packet under the MESSAGE OUT phase, since it is implemented as a special type of MESSAGE OUT. Thus you only see a small reduction in time for MESSAGE OUT, but no time spent in COMMAND phase. It is assumed that wide SCSI is in use, and the BCP information is sent at 160 MBytes per second. BCP speeds up the initial part of a transaction quite a bit, especially if multiple commands can be batched together.

The fourth column of Figure 2 shows the benefits of the Quick Arbitrate and Select (QAS) proposal. The QAS proposal eliminates the BUS FREE, ARBITRATION and SELECTION phases of standard SCSI and replaces them with a special QAS phase controlled by the current Target. QAS will revert to a standard BUS FREE phase if no other Devices are immediately in need of the SCSI bus, but in that case the bus is not fully utilized so there

is no need for QAS anyway. Where QAS will have the most impact is the heavily loaded bus case, and there it will be used nearly all the time. For that reason we have chosen to present results for the case where QAS is always successful at granting the bus to another device.

With QAS always succeeding, there is no SCSI BUS FREE phase, and significantly shorter times for the ARBITRATE and SELECT phases. The savings is proportionate to the number of disconnects, which are a function of the size of the transfer, the size of the disk buffers, and the buffer fill ratio.

The final three columns of Figure 2 show the overheads that result when all three enhancements are enabled. Separate overhead calculations are shown for the Initiator, Target and both, and can be directly compared to the numbers in Figure 1. Because both BCP and QAS try to reduce the cost of the initial arbitration of each transaction, their combined effect is less than the sum of their individual effects. However, the total savings is still quite impressive.

4 Overhead Savings for Typical Requests

Up to now the running example has been a read transaction with one Seek Disconnect and one Data Disconnect. The astute reader might want to know the savings for read transactions with other disconnection patterns or how those savings compare to the actual data transfer time. This section answers those questions by showing the overhead and data transfer times for four typical requests: a short sequential read, a long sequential read, a short random read and a long random read. All three of the overhead reduction proposals are included, plus a fourth proposal which adds a CRC to each block of data to improve robustness, at a slight increases in overhead. Figure 3 has the indicated times, plus an indication of the total time savings and the percentage of time that data (rather than protocol overhead) is occupying the SCSI bus.

As Figure 3 shows, overhead can consume a large fraction of bus bandwidth in the short read cases, but is significantly reduced with the proposed protocol enhancements. For sequential accesses, BCP has just as many arbitrations as normal SCSI, since you are trading N initiator arbitrations with some (or all data returned) with each transaction, for 1 initiator arbitration and N-1 target arbitrations for each N commands in a burst. Thus BCP’s only benefit is to reduce the time it takes to send the actual command and message out information. On Random requests, there would be seek disconnects anyway, so the savings in initiator arbitrations shows up as a real savings. Notice that for short Sequential reads BCP only increases data bandwidth from 57% to 59%, while on short Random reads data bandwidth is increased from 44% to 58%.

Protocol Enhancements:			SCSI-2	LFP Proposals (8 cmd burst, QAS "hit" fraction: 1)				
			Spec.	SMS	CRCA	BCP	QAS	All
Request type			**Min.**	**Min.**	**Min.**	**Min.**	**Min.**	**Min.**
Seq. Read 2K (no discon.)								
	data		12,800	12,800	12,900	12,800	12,800	12,900
	overhead		9,790	9,252	9,790	9,039	6,298	4,602
	total		**22,590**	**22,052**	**22,690**	**21,839**	**19,098**	**17,502**
	savings			538	-100	751	3,492	5,088
	% Usable BW		57%	58%	57%	59%	67%	74%
Seq. Read 64K (3 data disc.)								
	data		409,600	409,600	412,800	409,600	409,600	412,800
	overhead		31,420	30,468	31,420	30,669	17,542	15,432
	total		**441,020**	**440,068**	**444,220**	**440,269**	**427,142**	**428,232**
	savings			952	-3,200	751	13,878	12,788
	% Usable BW		93%	93%	93%	93%	96%	96%
Rand. Read 2k (1 seek disc.)								
	data		12,800	12,800	12,900	12,800	12,800	12,900
	overhead		16,462	15,924	16,462	9,106	9,508	4,670
	total		**29,262**	**28,724**	**29,362**	**21,906**	**22,308**	**17,570**
	savings			538	-100	7,356	6,954	11,693
	% Usable BW		44%	45%	44%	58%	57%	73%
Rand. Read 64k (1 S, 3 D disc.)								
	data		409,600	409,600	412,800	409,600	409,600	412,800
	overhead		38,092	37,140	38,092	30,736	20,752	15,500
	total		**447,692**	**446,740**	**450,892**	**440,336**	**430,352**	**428,300**
	savings			952	-3,200	7,356	17,340	19,393
	% Usable BW		91%	92%	92%	93%	95%	96%

Figure 3: Improvements for Selected Read Transactions

Long reads have considerably less overhead to begin with, as you would expect, but it certainly won't hurt to reduce it further. The results shown for long reads assume three data disconnects, though typical buffer full ratio operation will result in only one or two for even quite long transfers. However, the higher speed of transmission by Ultra might result in a few more disconnects, hence the assumption of three. More disconnects would increase the total overhead, and the benefit of the enhancement proposals.

In most cases QAS is more effective than BCP, since it reduces the time for all arbitration and selection operations, while BCP's only effect on arbitration and selection is to reduce the number of initiator arbitrations and selections. But, the effect of reduced initiator arbitrations and selections is evident in the short Random Read case, where BCP has a small advantage. Never-the-less, for best overall improvement you need both protocols.

The SMS proposal provides relatively little improvement compared to the other two, but is still a help. The final proposal, CRCA, whose benefit is improved data integrity, is shown to have negligible impact on performance.

While Figure 3 indicates significant improvement in available data bandwidth with the proposed protocol enhancements, the use of specification minimums as the basis for the calculations obscures the real magnitude of the benefits. While SCSI implementations have steadily approached the specification minimums over time, the current SCSI-2 implementations are still on the order of three to five times the theoretical minimums. To get an idea how much the LFP proposals would help in the current environment, Figure 4 shows the overheads and percentage of usable bandwidth assuming implementations which result in overheads four times the theoretical minimums. Note that under those circumstances, short random reads will only be passing real data 16% of the time with current protocols. But with LFP, the numbers grow to 41% for short requests and 87% for long!

				Overhead X 4				
Protocol Enhancements:			**SCSI-2**	**LFP Proposals (8 cmd burst, QAS "hit" fraction: 1)**				
			Spec.	**SMS**	**CRCA**	**BCP**	**QAS**	**All**
Request type			**Min.**	**Min.**	**Min.**	**Min.**	**Min.**	**Min.**
Seq. Read 2K (no discon.)								
	data		12,800	12,800	12,900	12,800	12,800	12,900
	overhead		39,160	37,008	39,160	36,155	25,192	18,409
	total		**51,960**	**49,808**	**52,060**	**48,955**	**37,992**	**31,309**
	savings			2,152	-100	3,005	13,968	20,651
	% Usable BW		25%	26%	25%	26%	34%	41%
Seq. Read 64K (3 data disc.)								
	data		409,600	409,600	412,800	409,600	409,600	412,800
	overhead		125,680	121,872	125,680	122,675	70,168	61,729
	total		**535,280**	**531,472**	**538,480**	**532,275**	**479,768**	**474,529**
	savings			3,808	-3,200	3,005	55,512	60,751
	% Usable BW		77%	77%	77%	77%	85%	87%
Rand. Read 2k (1 seek disc.)								
	data		12,800	12,800	12,900	12,800	12,800	12,900
	overhead		65,848	63,696	65,848	36,424	38,032	18,678
	total		**78,648**	**76,496**	**78,748**	**49,224**	**50,832**	**31,578**
	savings			2,152	-100	29,424	27,816	47,070
	% Usable BW		16%	17%	16%	26%	25%	41%
Rand. Read 64k (1 S, 3 D disc.)								
	data		409,600	409,600	412,800	409,600	409,600	412,800
	overhead		152,368	148,560	152,368	122,944	83,008	61,998
	total		**561,968**	**558,160**	**565,168**	**532,544**	**492,608**	**474,798**
	savings			3,808	-3,200	29,424	69,360	87,170
	% Usable BW		73%	73%	73%	77%	83%	87%

Figure 4: LFP benefits assuming implementations which are a factor of four worse than the theoretical minimums.