

Анализ разработанного модуля

Анализ критических путей в дизайне

Данный модуль является комбинационным, поэтому для анализа критических путей необходимо было добавить в проект файл, который будет содержать дополнительные параметры для его временного анализа. Был добавлен файл timing.xdc который содержал следующие строки:

- create_clock -name VCLK -period 5.0 -waveform {0 2.5} - создание виртуальных часов для подсчета временных задержек блока, с именем VCLK, периодом в 5 нс, переход с единицы на ноль происходит в середине периода.
- set_input_delay 0 -clock [get_clocks VCLK] [all_inputs] - для всех входных портов новый сигнал поступает через 0нс после VCLK нарастания.
- set_output_delay -clock [get_clocks VCLK] [all_outputs] - все выходные сигналы должны быть установлены за 0нс до VCLK нарастания.

При применении данных параметров я получил такой результат

Setup		Hold		Name	Waveform	Period (ns)	Frequency (MHz)
Worst Negative Slack (WNS):	0,003 ns	Worst Hold Slack (WHS):	1,606 ns	VCLK	{0.000 2.500}	5.000	200.000
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns				

Intra-Clock Paths - VCLK - Setup										
Name	Slack ¹	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	0.003	4	5	13	m_ready_i[1]	m_valid_o[1]	4.972	3.415	1.557	5.0
Path 2	0.020	4	5	32	rst_n	m_valid_o[0]	4.955	3.398	1.557	5.0
Path 3	1.100	2	3	3	s_valid_i[0]	s_ready_o[0]	3.875	3.292	0.584	5.0
Path 4	1.100	2	3	3	s_valid_i[1]	s_ready_o[1]	3.875	3.292	0.584	5.0

Intra-Clock Paths - VCLK - Hold										
Name	Slack ¹	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 5	1.606	2	3	3	s_valid_i[0]	s_ready_o[0]	1.631	1.385	0.246	0.0
Path 6	1.606	2	3	3	s_valid_i[1]	s_ready_o[1]	1.631	1.385	0.246	0.0
Path 7	1.880	3	4	32	rst_n	m_valid_o[0]	1.905	1.413	0.492	0.0
Path 8	1.881	3	4	32	rst_n	m_valid_o[1]	1.906	1.414	0.492	0.0

Судя по результатам критическими путями в данной архитектуре, являются выходные сигналы m_valid_o, хотя в коде они визуально должны иметь путь схожий с путем m_last_o. Судя по таблице основная задержка накапливается в "net delay". Обычно это значит, что расположение источника слишком далеко от расположения цели, или сигналы, от которых зависит результат, сильно разветвлены.

Рекомендации по изменению дизайна, которые могут повысить частоту и/или пропускную способность

Благодаря конвейеризации (временной параллелизм) можно увеличивает скорость работы схемы без сильного увеличения аппаратных затрат. Разделив данную схему на n участков, при условии, что временная задержка на всех участках одинаковая, мы сможем сохранить прежнюю латентность системы при серьезном повышении пропускной способности (частоты).

Также можно попробовать решить проблему с критическими путями, уменьшив задержку проводов, уменьшив их количество (возможно данную задержку можно уменьшить путем более плотного расположения схемы на ПЛИС)