2. Я реализовал конвейерную версию вычисления данной задачи. Это сделано для улучшения максимальной тактовой частоты схемы. Да это создает задержку в 3 цикла для первого вычисления, но при большом обьеме постоянных (валидных) обрабатываемых сигналов средняя латентность близка к 1 циклу. Это очень близка латентноси в не конвейерной реализации схемы где латентность составляет ровно 1 цикл. Но при этом максимальная тактовая частота хуже в разы. И средняя латентность по времени (в ns) будет лучше.

Так же для оптимизации используемых ресурсов (меньше DSP) и для уменшения задержки комбинационной логики я использовал arithmetic left shift by 2 для умножения на 4, и arithmetic right shift by 1 для деления на 2.

11. В целом существует много метотодов.

- Я в своем коде использовал самый простой способ где разрядность выходных и внутренних сигналов всегда гарантирует что не будет переполнения. То есть сигналы разширены до необходимой разрядности. Например для промежуточного результата “1 + 3c” я использовал сигнал разрядностью WIDTH + 3, где WIDTH это разрядность входных сигналов. WIDTH + 3, потому что для умножения на 3 без переполнения нужно разширить сигнал на 2 разряда, и для +1 нужно разширить ещё на один разряд.
- Можно использовать флаги переполнения для сигнализации ошибки, с или без последующей обработкой ошибки. Последующая обработка может включать в себя ипользование насыщающей арифметики, где после обнаружения переполнения, значение результата заменяется на максимальное или минимальное значение в зависимости от знака выходного сигнала.

12. Аппаратные ресурсы при реализация в Vivado на target device-е “xc7z035fbg676-1”, и при разрядности входных параметров 16 бит указано ниже:

- Slice LUTs: 34
- Slice Registers: 19
- DSPs: 1

Name 1	Slice LUTs (171900)	Slice Registers (343800)	Slice (54650)	LUT as Logic (171900)	DSPs (900)	Bonded IOB (250)	BUFGCTRL (32)
N math_equation	34	19	20	34	1	102	1

13. Я сначала ограничил clk на 500 MHz (2 ns) и получил WNS = 0.913 ns. То есть теоретический система может работать на 919 MHz.

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.913 ns	Worst Hold Slack (WHS):	0.072 ns	Worst Pulse Width Slack (WPWS):	0.175 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	51	Total Number of Endpoints:	51	Total Number of Endpoints:	21

All user specified timing constraints are met.

Но когда ограничиваю систему на 833.33 MHz (1.2 ns) WNS, THS всё ещё положительны, но Total Pulse Width Negative Slack (TPWS) получается негативным, то есть временные ограничения не соблуденены.

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.039 ns	Worst Hold Slack (WHS):	0.077 ns	Worst Pulse Width Slack (WPWS):	-0.625 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	-1.025 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	2
Total Number of Endpoints:	51	Total Number of Endpoints:	51	Total Number of Endpoints:	21

Timing constraints are not met.

Я никогда до этого не работал с TPWS, поэтому менял временные ограничения мануально и нашел примерную максимальную тактовую частоту работы схемы при котором временные ограничения соблудаются. Оно примерно равно 545.54 MHz (1.85 ns).

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.670 ns	Worst Hold Slack (WHS):	0.072 ns	Worst Pulse Width Slack (WPWS):	0.025 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	51	Total Number of Endpoints:	51	Total Number of Endpoints:	21

All user specified timing constraints are met.

Также я не учель input and output delays в ограничениях, и из-за этого максимальная тактовая частота немного отличается от реальных значении, и считается более оптимистическим. В реальности как минимум сушествует задержка из-за routing-а, но также существуют задержки чтения из памяти/регистров.