A High-level Synthesis Design Flow from ESL to RTL

Dr. Vipul Kumar Mishra (Assistant Professor)

Department of Computer Science Engineering

Bennett University, Greater Noida

Introduction

High-level synthesis (HLS) acts as a connecting link from the behavioural-level specification at the electronic system level (ESL) to the structural building block at the register transfer level (RTL).

Design Flow [1]

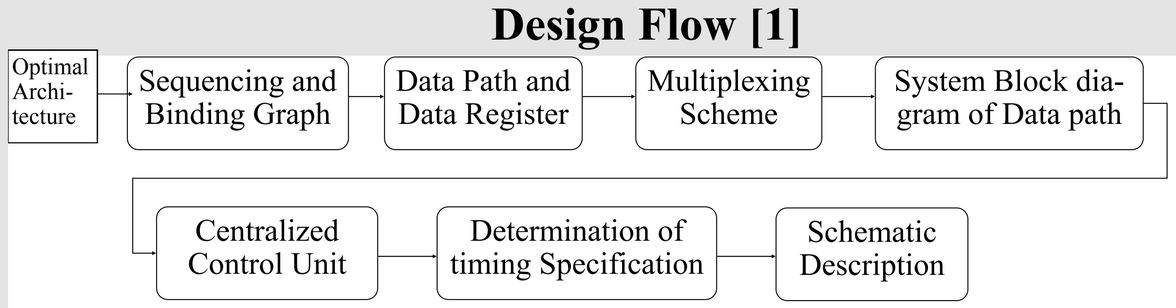

Fig. 1 Design flow

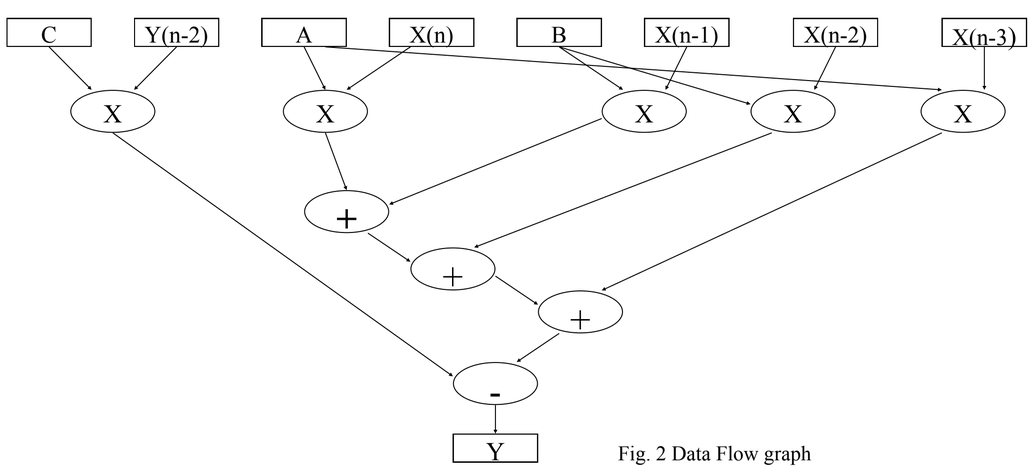

Fig. 2 Data Flow graph

Phase 1 Scheduling

After finding the optimal architecture we perform scheduling. Scheduling is a process that states the time slot for every operation while fixing the timing length (latency) in such a manner so that the synthesized hardware structure meet the timing restriction.

Phase 2 Sequencing and Binding graph

Representation of the resources in the temporal and spatial domains is performed with the aid of a sequencing and binding graph. Sequencing graphs are used to specify the nature of operation, i.e. at which time what resources are subjected to which operation. The graph is a direct reflection of the circuit for the particular application used. Binding the resources in the sequencing graph of helps to formalize a methodology of incorporating the multiplexers and demultiplexers into the circuit, and is necessary for system design [2].

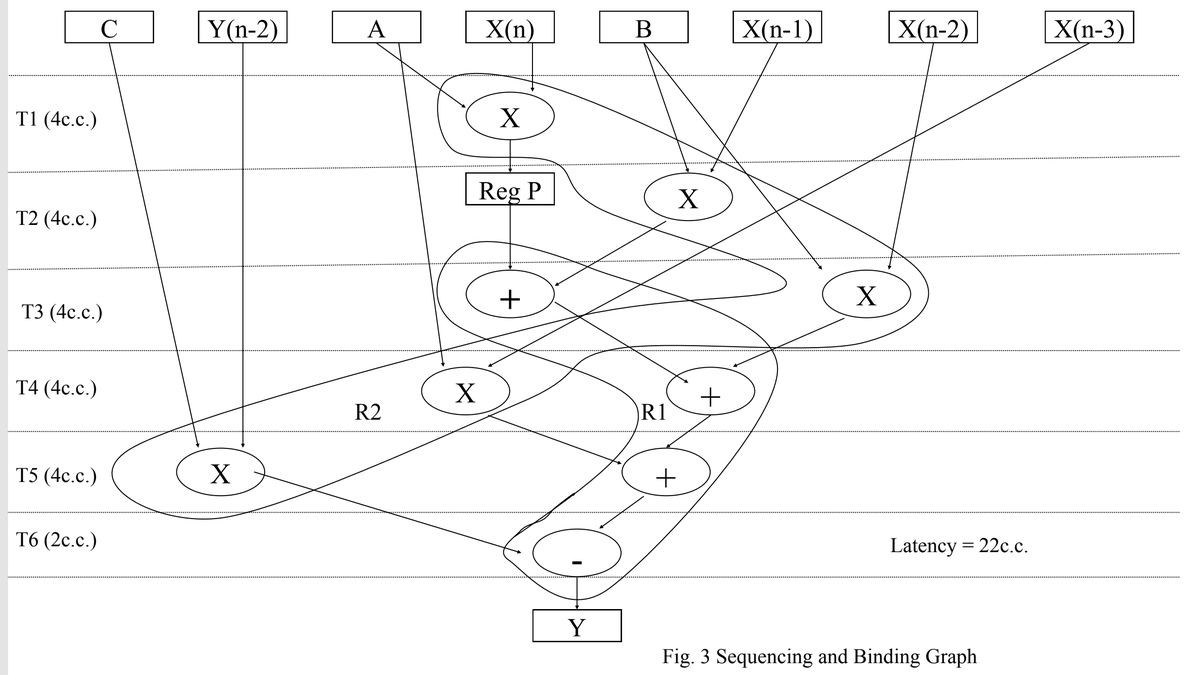

Fig. 3 Sequencing and Binding Graph

Phase 3 Determination of multiplexing scheme

A multiplexing scheme is a way of representing each resource of the system with its respective inputs, outputs, operations and the necessary interconnections.

Table 1 Multiplexing scheme for Adder/Subtractor

Time	Operation	Input 1	Input 2	Output
0	-	-	-	-
1	-	-	-	-
2	-	R2out	Reg P	-
3	+	R2out	R1 out	R1 in
4	+	R2out	R1 out	R1 in
5	+	R2out	R1 out	R1 in
6	-	-	-	Reg Y
7	-	-	-	-

Table 1 Multiplexing scheme for Multiplier

Time	Operation	Input 1	Input 2	Output
0	-	Reg A	Reg x(n)	-
1	*	Reg x(n-1)	Reg B	Reg P
2	*	Reg x(n-2)	Reg B	R1 in
3	*	Reg A	Reg x(n-3)	R1 in
4	*	Reg C	Reg (n-2)	R1 in
5	*	-	-	R1 in
6	-	-	-	-
7	-	-	-	-

Phase 4 Development of the System block diagram

The system block diagram consists of two divisions: data path and control path. The data path is responsible for the flow of data through the busses and wires after the operations have been performed by the components present in the data path circuit.

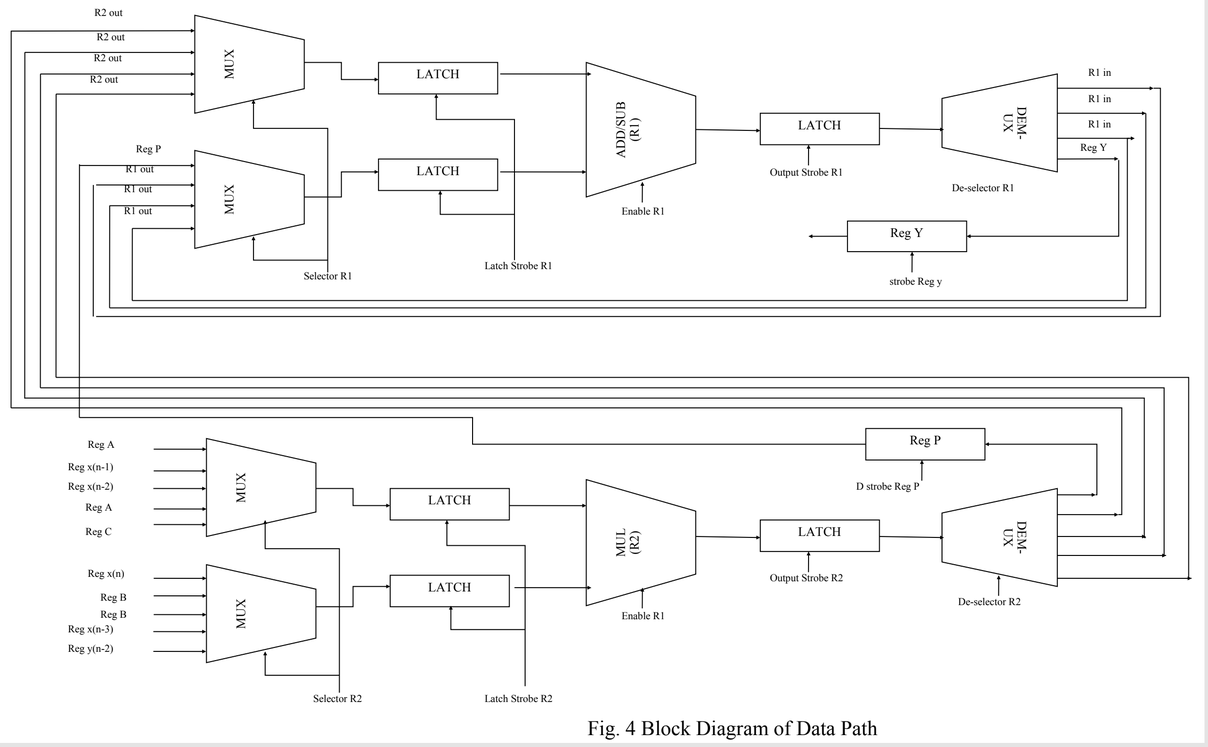

Fig. 4 Block Diagram of Data Path

Phase 5 Development of centralized control unit with timing specification

The function of the controller is to activate and deactivate the different elements of the data path based on the timing specification determined for the objective function. For synchronous functioning of all data elements in the system the controller must respond to the requirement at exactly the right moment. The determination of the timing specification from control unit helps to create an accurate structure of the controller.

Table 3 Timing specification for the data path circuit

Adder(R1)						Multiplier(R2)					Strobes			
Latch Strobe	Enable	Add_sub	Output strobe	Selector	Deselector	Latch Strobe	Enable	Output strobe	Selector	Deselector	Strob_reg	Dstrob_regP	Strobe_rgY	Clock Cycle
0	0	0	0	00	0	0	0	0	000	000	0	0	0	0
									000		1			1
						1								2
							1		001	000				3
						0								4
														5
								1						6
								1						7
							0					1		8
						1								9
				00			1	0	010	001				10
						0								11
														12
														13
							0	1						14
1						1								15
	1	0		01	00		1	0	011	010				16
														17
0						0								18
														19
	0		1				0	1						20
1						1								21
	1	0	0	11	10		1	0	100	011				22
0						0								23
														24
														25
	0		1				0	1						26
1						1								27
	1	0	0	11	10		1	0		100				28
0						0								29
														30
														31
	0		1				0	1						32
1														33
	1	1	0		11									34
														35
0			1											36
	0												1	37

Phase 6 Development of schematic structure for the whole

First, all the components in the data path were implemented in VHDL before system simulation. Then, the schematic structure of the whole system was designed and implemented in the Xilinx Integrated Software Environment (ISE).

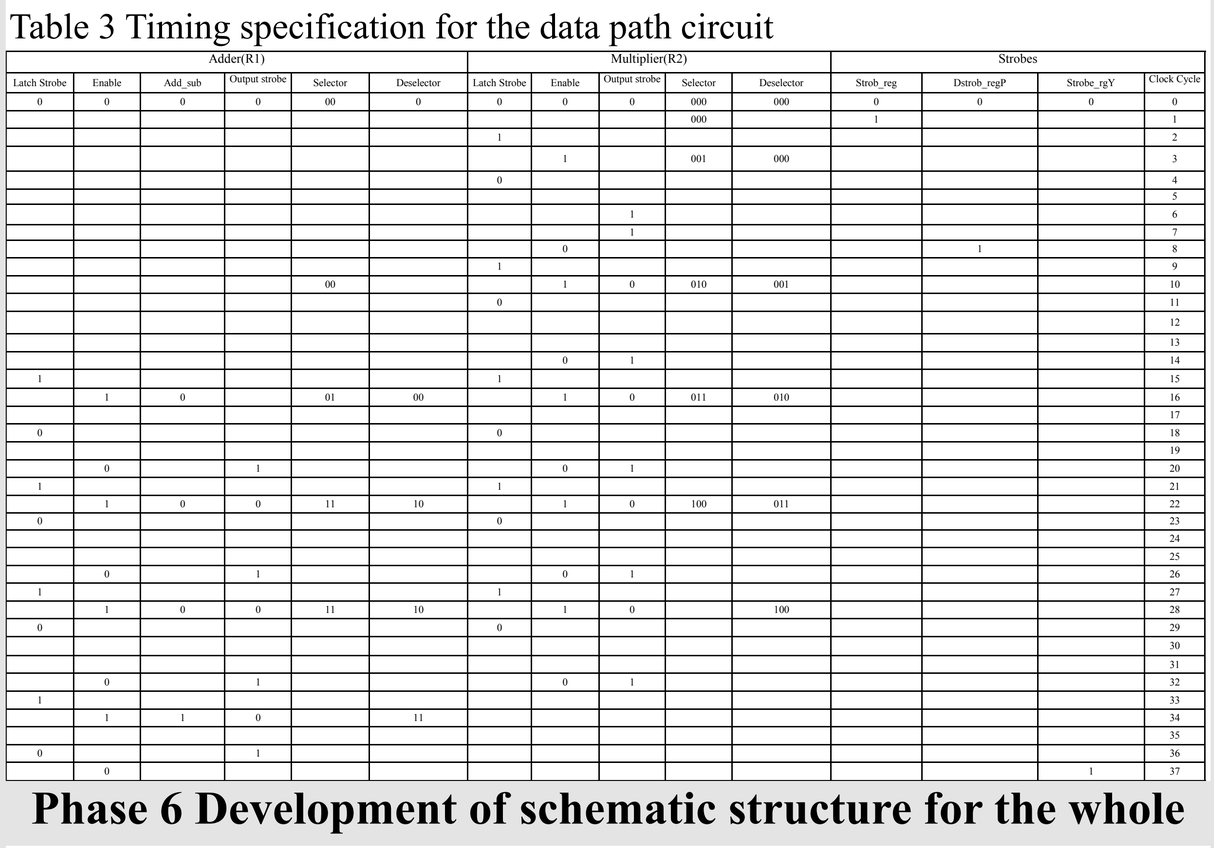

Fig. 5 Schematic view of the system