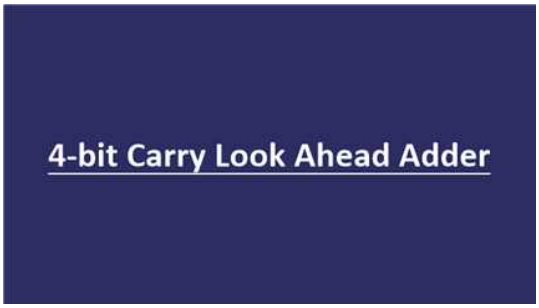
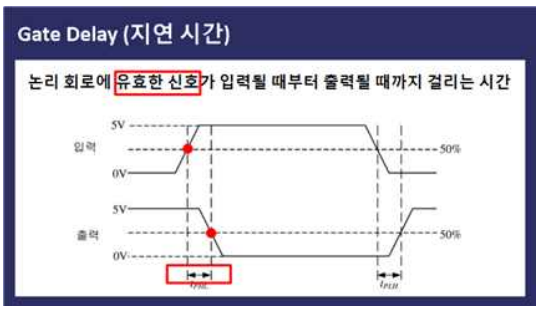
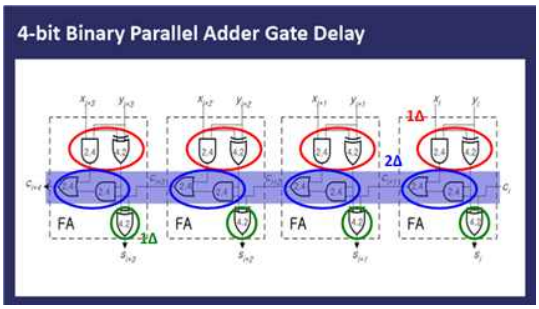
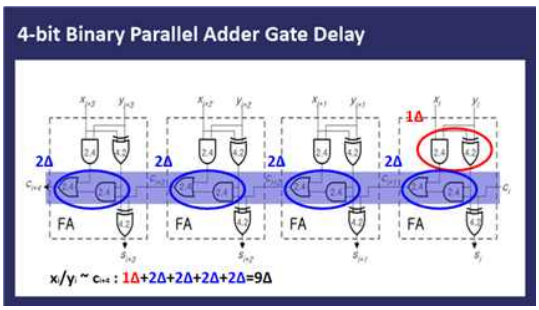
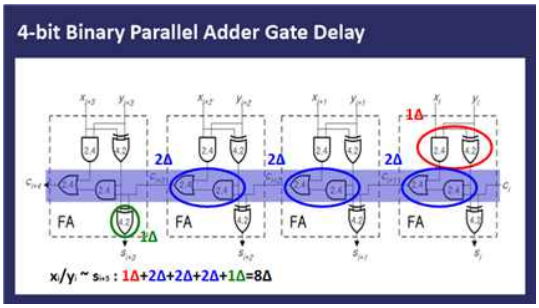
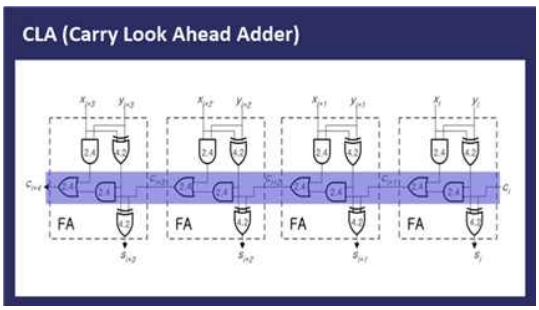
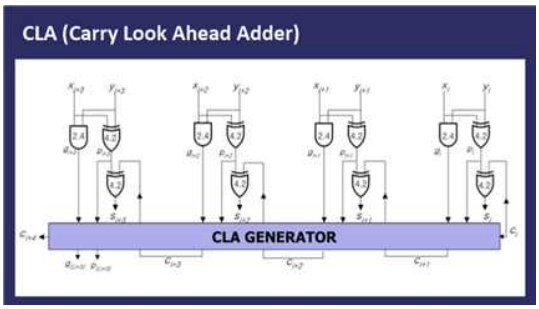
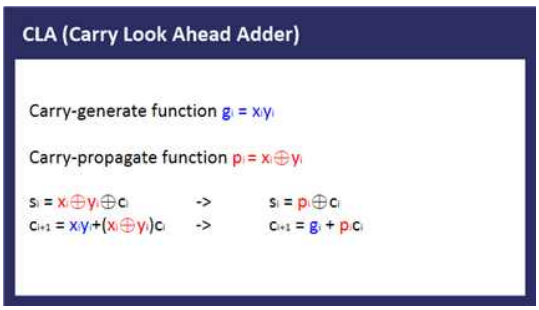
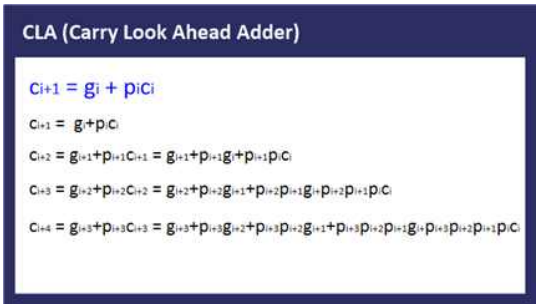
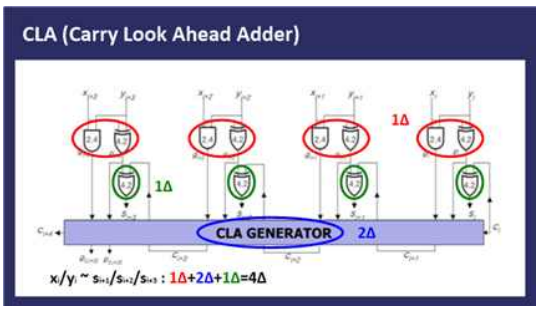
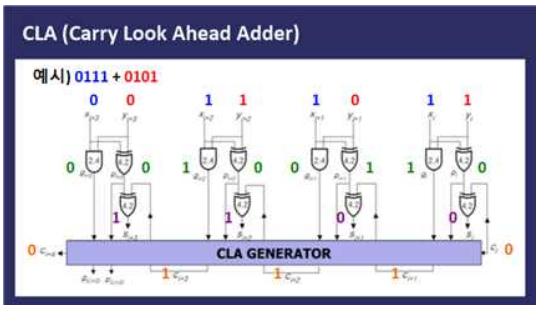
	4-bit carry look ahead adder 부분 설명을 맡은 김기철입니다.
	CLA에 대해 설명하기에 앞서 Gate delay, 지연 시간에 대해 설명하겠습니다. 지연 시간이란 논리 회로에 유효한 신호가 입력될 때부터 출력될 때까지 걸리는 시간입니다. 여기서 중요한 부분이 이 유효한 신호인데, 신호가 임계치(50%)를 넘어섰을 때부터를 유효한 신호라고 합니다. 따라서, 단순히 입력 신호가 들어왔을 때 바로 출력이 되는 것이 아니라 입력 신호가 유효한 이 시점부터 출력 신호가 유효한 이 시점까지의 시간을 지연 시간이라고 합니다. 이 지연 시간이 작을수록 데이터를 빠르게 처리하고 전반적인 시스템이 향상된다는 장점이 있습니다.
	앞서 설명한 4개의 FA로 이루어진 병렬 가산기의 지연 시간을 구해보겠습니다. 병렬 가산기는 AND, OR XOR gate로 구성되어 있는데, 실제로 이 게이트들은 모두 다른 지연 시간을 가지고 있습니다. 하지만 계산을 할 때, 임의로 모든 게이트의 지연 시간이 1Δ라고 가정하겠습니다. 따라서 처음 x, y가 입력되었을 때, carry와 s를 위한 AND와 XOR 연산을 하는 이 과정에서 1Δ가 소요됩니다. 그 다음 위 연산의 결과를 통해 다음 자리에 넘겨줄 carry를 구해주는 이 과정에서 2Δ가 소요되고, 마지막 각각 s를 구해주는 이 과정에서 1Δ가 소요됩니다.
	이 정보를 가지고 직접 4-bit 병렬 가산기의 지연 시간을 구해보겠습니다. 먼저 xi/yi부터 ci+4를 구할 때는 이 과정을 차례대로 연산해야 하므로 각각의 지연 시간을 모두 더한 9Δ가 소요됩니다.

	비슷하게 xi/yi부터 si+3을 구할 때는 이 과정을 차례대로 연산해야 하므로 각각의 지연 시간을 모두 더한 8Δ가 소요됩니다.
	이렇게 4-bit 병렬 가산기의 지연 시간을 구해보았는데, 이 병렬 가산기는 지연 시간이 크다는 단점이 있습니다. 지연 시간에 가장 큰 영향을 주는 것은 이 carry 연산을 하는 부분인데, ci+4를 구하기 위해서는 이전의 carry값인 ci+3, ci+2, ci+1의 값을 모두 알아야 하기 때문입니다. 따라서 이를 해결하는 방법인 CLA에 대해 설명하겠습니다.
	4-bit CLA adder의 논리 회로는 다음과 같습니다. 병렬 가산기의 지연 시간에 큰 영향을 주었던 carry 연산 부분에 CLA generator를 대체하여 지연 시간을 낮출 수 있습니다.
	CLA generator 연산을 위해 다음과 같은 g와 p를 만들었습니다. g는 carry-generate function으로 기존 연산과 관계없이 반드시 자리올림수가 생성됨을 확인하는 값이고 gi=xiyi로 표현합니다. p는 carry-propagate function으로 추가적으로 자리올림수가 발생할 가능성을 검사하는 값이고 pi=xi⊕yi로 표현합니다. 이렇게 표현한 g와 p로 다음과 같이 새롭게 s와 c에 대한 식을 표현할 수 있게 됩니다.

	여기서 c를 구해주는 이 연산을 보면, 처음 입력으로 주어지는 ci를 통해 다음과 같이 재귀적으로 ci+1부터 ci+4까지 한 번에 구할 수 있게 됩니다. 이는 기존의 병렬 가산기처럼 이전의 carry값을 기다릴 필요가 없다는 것을 의미하고, ci+1=gi+pici이므로 carry값을 구할 때 AND gate 한 개, OR gate 한 개가 필요하므로 2Δ만에 모든 carry 값을 구할 수 있게 됩니다.
	이러한 CLA 연산을 이용하여 4-bit CLA adder의 지연 시간을 구해보겠습니다. 처음 x, y가 입력되었을 때, p와 g를 구하는 이 과정에서 1Δ가 소요되고, 입력된 ci로부터 CLA generator에서 모든 carry값을 구하는 이 과정에서 2Δ가 소요되고, 각 자리의 s값을 구하는 이 과정에서 1Δ가 소요됩니다. 따라서 xi/yi에서 si+1/si+2/si+3을 구하는데 4Δ가 소요됩니다. 병렬 가산기에 비해 지연 시간이 작아진 것을 확인할 수 있고 더 효율적으로 연산을 할 수 있게 됩니다.
	병렬 가산기와 같은 예시로 다음 계산을 해보겠습니다. 먼저 각 자리의 x와 y가 입력이 되면 각 자리의 p와 g를 다음과 같이 구할 수 있습니다. 그 다음 입력된 ci로부터 CLA generator 연산으로 ci+1부터 ci+3까지 모든 carry 값을 다음과 같이 구할 수 있습니다. 마지막으로 각 자리의 s값을 다음과 같이 구하여 최종 덧셈결과가 1100, 최종 carry값이 0인 것을 확인할 수 있습니다.
	이상입니다.