ELGS 기법을 이용한 FLEX 디코더용 심볼 동기회로 설계†

이태응, 강민섭
안양대학교 컴퓨터학과
tylee@cs2.anyang.ac.kr, (phone) 0343-467-0876

Design of Symbol Synchronizer for FLEX Decoder Based on ELGS Technique

Tae Eung Lee, Min Sup Kang
Dept. of Computer Science, Anyang University
tylee@cs2.anyang.ac.kr, (phone) 0343-467-0876

요 약

본 논문은 FLEX 디코더에서 필요한 심볼 클럭을 생성하기 위한 심볼 동기 알고리즘을 제안하고, 제안한 알고리즘을 기본으로 한 심볼 동기회로의 설계에 관한 것이다. 제안한 알고리즘은 조-만 게이트 동기 (ELGS: Early-Late Gate Synchronization)기법을 이용하고 있다.

VHDL(VHSIC Hardware Description Language)로 설계된 심볼 동기회로는 Synopsys 툴을 이용하여 기능 레벨의 시뮬레이션을 수행하였고, Altera MAX+plus II 를 이용하여 타이밍 분석을 수행하였다. 실험 결과로부터 Source unit와 FLEX 디코더와의 시스템 동기가 정확히 이루어짐을 확인하였다.

I. 서론

송신기에서 신송되는 신호를 수신하기 위해서는 신호를 수신할 시스템에서 동기가 이루어져야한다.

시스템 동기화는 크게 반송파 동기화, 심볼(비트) 동기화 그리고 패킷 동기화의 세 가지 형태로 구분된다[1]. 반송파 동기화는 위상 동기 시스템의 복조 과정에서 필요한 반송파의 위상과 주파수를 추정하기 위해 반송파 억압(carrier-suppressed) 파형으로부터 반송파를 복원하는 방법이다. 심볼 동기화는 기저대역 데이타 심볼 열(stream)과 수신기의 클럭을 동기 시키는 방법이다.

†본 연구는 정보통신부의 정보통신 연구개발사업의 일환으로 추진된 공동 기술개발사업임.

또한, 반도체설계교육센터(IDEC) 사업의 지원을 받아 수행되었음.

마지막으로 패킷 동기화는 동기를 목적으로 데이터 열에 반복적인 비트나 문자를 삽입하는 특별한 메시지 형태를 이용하는 동기화 방법이다[2,3,5].

본 논문에서는 상기한 세가지의 동기화 방법 중에서 심볼(비트) 동기화 방법을 이용한다. 심볼 동기화는 앞에서 언급한 것처럼 전송된 정보(데이타)를 복원하기 위해서 정합여과기(matched filter)나 상관기(correlator)의 병렬군에 대한 출력을 주기적으로 표본화하는 클럭 주파수를 결정짓는 방법으로서, ELGS가 이에 속한다.

본 논문은 ELGS 기법을 이용하여 FLEX 디코더[4]에서 필요한 심볼 클럭을 생성하기 위한 심볼 동기회로의 설계 및 구현에 관한 것이다.

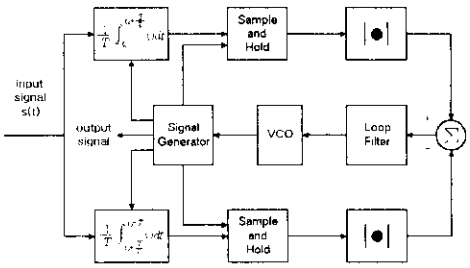
심볼 동기회로는 Synopsys 상에서 VHDL을 이용하여 기능 설계 및 시뮬레이션을 수행하였고, Altera MAX+plus II를 이용하여 타이밍 분석을 수행하였다. 실험 결과로부터 Source unit와 FLEX 디코더와의 시스템 동기가 정확히 이루어짐을 확인하였다.

II. 조-만 게이트 동기 회로

조-만 게이트 동기 회로(ELGS: Early-Late Gate Synchronizer)는 data rate 1/T를 비교하는 비교적 좁은 대역폭(Bandwidth)을 가지는 기본적인 폐-루프 제어(closed-loop control)시스템이다. 조-만 게이트 동기 회로의 기본적인 형태는 다음과 같다[1,2,3,5].

조-만 게이트 동기 회로는 정합 여파기나 상호 상관기를 사용하여 구현되며, (그림 1)에서 조-만 게이트 동기 회로의 블럭도를 나타내고 있다.

입력 신호 s(t), $0 \le t \le T$, 가 구형파라 할 때 Early-Late Gate인 두 개의 적분기는 심볼 간격 T상에



(그림 1) 조-만 게이트 동기 회로의 블럭도

서 적분한다. Early Gate의 적분기는 추정된 최적 표본화 시간에 관련된 초기의 δ초에서 적분을 시작하고, Late Gate의 적분기는 추정된 최적 표본화 시간에 관련된 후기의 δ초에서 적분을 시작한다. Sample and Hold의 출력은 각 적분간격 동안 이전의 적분값을 유지하여 계단 모양을 갖는 원래 신호의 근사값을 각 적분기의 절대값 블럭으로 전달한다. 오차 신호는 두 적분기의 절대값 출력 사이에서 차이를 가짐으로써 발생하며, 서로 대칭이므로 두 적분기 사이의 차이인 오차 신호는 zero가 되어야 한다. 동기가 맞지 않아 오차 신호가 zero가 아닐 경우에는 오차 신호가 zero이 되도록 시간상으로 오차 신호의 크기에 따라서 앞, 혹은 뒤로 이동된다. 그리고 저역 통과 여파기(Low Pass Filter)를 사용하여 신호 표본을 방해하는 잡음에 대한 오차 신호를 완만하게 한다[3]. 저역 통과 여파기의 출력은 전압 제어 발생기(VCO : Voltage-Contr olled Oscillator)의 제어 신호가 되며, 전압 제어 발생기의 출력은 주기적으로 신호 발생기를 구동시킨다. 신호 펄스 발생기의 출력이 입력된 신호에 대한 동기 신호이며, 신호 펄스 발생기의 출력은 다시 적분기로 되돌아간다.

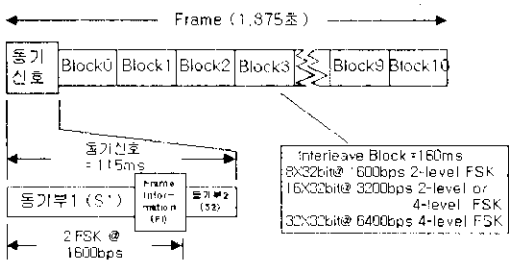
III. 심볼 동기회로의 설계

이 장에서는 고속 Pager를 위한 FLEX 프로토콜 및 입력신호의 구성을 기술하며, 심볼 동기회로를 설계하기 위한 심볼 동기 알고리즘을 기술한다.

3.1 FLEX 프로토콜

미국의 모토롤라사에 의해서 제안된 FLEX 프로토콜은 고속 호출기용으로 널리 이용되고 있다[4]. 이 프로토콜은 1,600, 3,200 및 6,400 bps의 세 가지 전송속도를 지원한다. (그림 2)는 FLEX 프레임에 대한 기본 구성을 나타낸다.

FLEX 프로토콜은 15개의 cycle로 구성되며, 각 Cycle은 128개의 Frame으로 구성된다. (그림 2)는 FLEX 프레임의 기본 구성을 나타낸다. 각 프레임은 동기신호부와 11개의 Block으로 구성되므로 Block에는 0부터 10까지의 번호가 부여된다. 전송속도가 1600 bps의 경우는



(그림 2) FLEX 프레임의 기본 구성

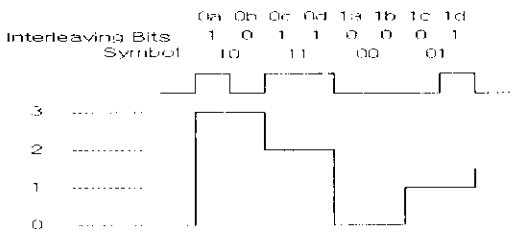
1 Block은 8개의 Word로 구성되며 11개의 Block에는 88개의 Word가 존재한다. 3200 bps 및 6400 bps의 전송속도의 경우는 Phase multiplexing이 행하여지는데, 이 경우는 각 Phase에 대하여 88개의 Word로 구성된다.

동기신호부는 Sync1, FI(Frame Information), Sync2로 구성되는데 FI에는 4-bit로 표시되는 Cycle 번호와 7-bit로 표시되는 프레임 번호를 포함한다. 각 프레임의 Sync1(S1)은 1600 bps로 송신되는데 프레임의 timing, 1600 bps 심볼의 timing 및 frame 이외의 부분에 송신되는 신호의 속도에 대한 정보를 알려준다. 각 프레임의 Sync2(S2)는 FI에서 알려준 block의 속도에 따라 동기되어 입력되는데, message block의 다중화 신호를 정확하게 분리하여 Decoding하기 위한 의도도 있다.

3.2 입력 신호의 구성

본 논문에서 설계한 심볼 동기회로의 입력은 RF front end로부터 입력된 4-level audio 신호(그림 3의 레벨 0, 1, 2, 3)를 2-bit 아날로그-디지탈 변환기(ADC)에 의해서 Interleaving된 비트(심볼)로 변환되어진 2-level의 디지탈 신호는 실제 FLEX Pager에 사용되는 입력단자인 EXTS0와 EXTS1로 인가된다.

(그림 3)는 4-level 신호 및 Interleaving 비트 심볼을 나타내며, Interleaving된 데이타 비트는 FLEX Pager System의 동기 신호부의 입력으로 사용된다.



(그림 3) 4-level 신호 및 Interleaving 비트 심볼

3.3 심볼 동기 알고리즘

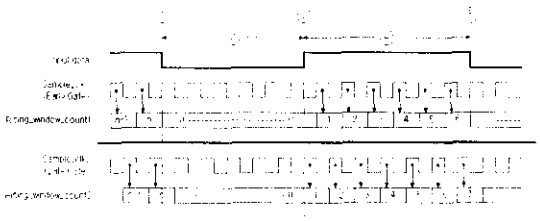
제안된 심볼 동기기법은 조-만 게이트 동기 알고리즘

을 기본으로 하고 있으며[1,3], 기본적인 원리는 ADC를 통해 전송되어진 '0' 과 '1'의 심볼 주기가 유지되는 동안 Sample_clk1과 Sample_clk2를 이용하여 각각 적분한 결과값을 비교하는 것이다. Early/Late gate accumulator에서 적분을 수행하기 위해 각각 Sample_clk1과 clk2를 사용하게 된다. 심볼 동기회로의 입력신호는 A/D 변환기를 통해 2-level로 변환된 디지탈 신호이며, 동기회로의 입력단인 EXTS0와 EXTS1으로 인가된다. 제안된 알고리즘은 다음과 같이 5단계에 의해서 수행된다.

[단계 1] 입력되는 '0', '1'의 심볼들과 영역에 대한 비교 및 적분을 수행하기 위하여 2개의 Sample clock을 생성한다. 이 두 클럭은 각각 Early Gate와 Late Gate에 대응된다. 즉, Main clock(76.8 KHz)을 2분주한 신호인 2개의 Sample clock(38.4 KHz)을 각각 Sample_clk1과 Sample_clk2라 정의한다.

[단계 2] 위상 오류가 위상 한계(최대 4bit)를 초과하는지를 check하기 위해서 2개의 Sample clock들을 이용하여 적분값을 서로 비교한다.

[단계 3] Sample_clk1과 Sample_clk2를 이용하여 입력 데이타가 각각 '1'과 '0'인 적분값을 비교할 때 발생가능한 4가지 경우를 체크한다(그림 4, 5).



(그림 4) 데이터가 1인 경우 적분값 비교

(그림 5) 데이터가 0인 경우 적분값 비교

[경우 1] Sample_clk1과 Sample_clk2 사이에 심볼이 전송되고, 심볼의 event가 이 두 클럭 가운데에서 발생했을 경우 적분값이 같으면, 타이밍 교정이 필요 없다((그림 4, 5)의 ①).

[경우 2] 각 Sample_clk1과 Sample_clk2사이에 동일한 심볼이 전송될 경우, 적분값은 동일하다. 이것은 2개의 클럭안에서 심볼 변화가 발생되지 않은 것을 의미한다. 이 조건에서는 타이밍 교정이 필요 없다((그림 4, 5)의 ④).

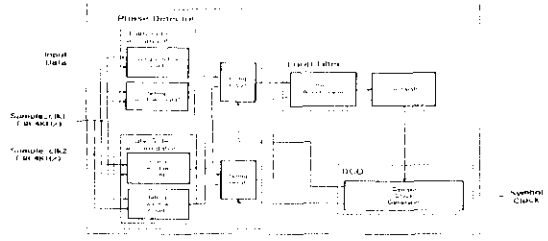
[경우 3] Sample_clk1에서의 적분값이 Sample_clk2에서의 적분값 보다 클 경우, 심볼 주기의 Sample_clk2에서 심볼 변화가 있다는 것을 의미한다. 이 경우 심볼 영역 외 동기를 맞추기 위해 타이밍 교정이 필요하다((그림 5)의 ⑤).

[경우 4] Sample_clk2에서의 적분값이 Sample_clk1에서의 적분값 보다 클 경우, 심볼 주기의 Sample_clk1에서 심볼 변화가 있다는 것을 의미한다. 이 경우 심볼 영역의 동기를 맞추기 위해 타이밍 교정이 필요하다((그림 4)의 ⑤).

[단계 4] 비교된 결과에 대해서 위상차를 조절하고, 오류 체크를 행하여 심볼 클럭을 생성한다.

3.4 심볼 동기회로의 설계

(그림 6)은 제안된 알고리즘을 이용한 심볼 동기회로의 블럭도이다. 심볼 동기회로는 위상 검출기(Phase Detector), Loop Filter, 그리고 Digitally Controlled Oscillator(DCO)를 포함한다.



(그림 6) 심볼 동기회로의 블럭도

(1) 위상 검출기

위상 검출기는 Early/Late Gate인 두 개의 적분기 설계를 말한다. EXTS0와 EXTS1단자에서 출력되는 입력 신호와 마스터 클럭(76.8KHz)을 2분주한 2개의 Sample_clk1(38.4KHz)과 Sample_clk2(38.4KHz)를 받아들여 Early/Late Gate Accumulator의 rising window count1과 count2에서 1의 상태에 대한 적분을 수행하고, falling window count1과 count2는 0의 상태에 대한 적분을 수행한다. 적분된 결과값은 각각 rising과 falling result에 입력되어 심볼 클럭과 데이터 열 사이의 위상차와 오류를 출력한다. Early와 Late Gate 적분기 사이의 차는 위상 오차(θ_e)와 비례한다. 위상 검출기의 출력은 θ_e $|\Sigma_{early}| - |\Sigma_{late}|$ 와 같이 정의된다.

(2) Loop Filter

Loop Filter의 Error Accumulator는 위상 검출기로부

터의 위상 오류(θ_e)의 출력이 누적된다. Early/Late Gate에서 발생하는 위상 오류의 갯수가 동일할 경우 DCO에서 발생되는 심볼 클럭에는 아무런 변화가 없으며, 양단의 위상 오류 발생 갯수의 차(difference)만이 심볼 클럭에 대한 위상 오류의 원인으로 작용하게 된다. Comparator는 위상 오류가 위상 한계(최대 4bit)를 초과하는지를 check한다.

(3) Digitally Controlled Oscillator(DCO)

DCO는 위상 오류가 위상 한계를 초과했을 때, 내부에서 발생된 심볼 클럭의 위상을 조절하며, DCO의 출력은 다시 Phase Detector의 rising과 falling result에 입력되어 심볼 Clock과 데이터 열 사이의 위상 차와 오류를 조절한다.

DCO에서 생성된 심볼 클럭의 출력은 2-bit ADC내에 있는 Peak-and-Valley DAC에 사용되는 Counter을 구동시킨다. ADC는 입력신호로부터의 잡음을 제거하기 위한 3차 Butterworth 스위칭-캐패시터 필터를 사용해 1KHz와 2KHz의 cutoff 주파수를 검출하고 3개의 스위칭-캐패시터 비교기들을 사용하여 2-bit ADC에 최대·최소 전압(Vref+와 Vref-)을 제공하는 Filter의 출력 신호인 Peak-and-Valley DAC에 의해 4-level audio 신호가 2-level 디지털 신호로 변환된다[4].

IV. 실험 결과

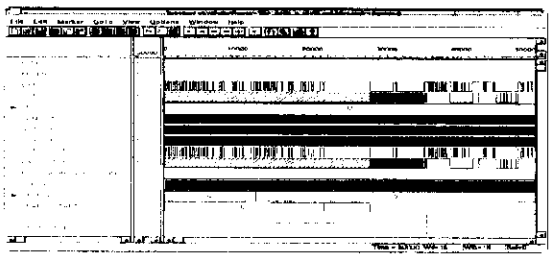
본 논문에서는 FLEX 방식을 사용하여 외부에서 수신된 data와 FLEX Pager System과 정확한 동기화를 이루기 위한 심볼 동기회로를 설계하였다. 설계된 심볼 동기회로는 VHDL을 이용하여 기술하였고, front-end 설계는 워크스테이션 상에서 Synopsys 툴을 이용하였으며, Back-end 설계는 Alera사에서 제공하는 MAX+plus II를 이용하였다. (그림 7)은 SYNOPSYS^{TM}의 VHDL 시뮬레이터를 이용하여 6400 bps/4-level에 따라 동작하는 기능 레벨 시뮬레이션을 수행한 결과를 나타낸다.

(그림 8)과 (그림 9)는 각각 Altera의 타이밍 시뮬레이터를 이용하여 각 bps에 대한 시뮬레이션 결과와 Logic Analyzer를 이용한 실험 결과를 나타내었다.

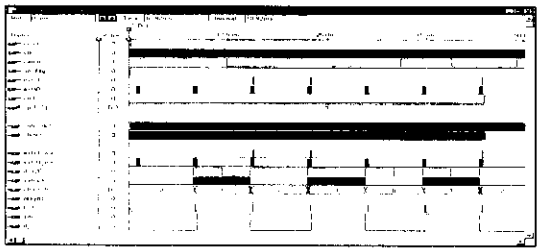
V. 결론

본 논문에서는 FLEX 디코더에서 필요한 심볼 클럭을 생성하기 위한 심볼 동기 알고리즘을 제안하고, 이를 이용한 심볼 동기회로의 설계 및 구현에 관하여 기술하였다. 제안된 심볼 동기회로는 FLEX pager에서 요구하는 최고 6400 bps의 통신속도를 가지며, 설계가 용이한 장점을 갖는다.

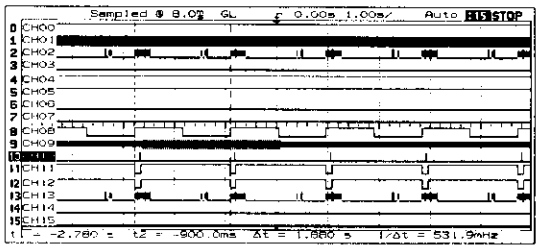
설계된 각 모듈에 대한 VHDL 모델링 및 논리합성은 Synopsys^{TM}툴을 이용하였고, Altera MAX+PLUS II 상에서 Test_Vector를 구성한 후 각각의 1600, 3200, 6400 bps에 따른 동작상태는 타이밍 시뮬레이션 결과로부터 확인하였다. 또한, 설계된 심볼 동기회로를 테스트 보드상에서 데이터를 down-load한 후 Logic Analyzer를 이용하여 칩의 동작상태를 확인하였다.



(그림 7) 6400 bps/4-level 기능 시뮬레이션 결과



(그림 8) 각 bps에 대한 타이밍 시뮬레이션 결과



(그림 9) Logic Analyzer를 이용한 실험 결과

[참고문헌]

[1] John G. Proakis "Digital Communications," Third Edition, vol 6, pp. 333-371, 1995.

[2] F.M. Gardner, Phase Lock Techniques, 2nd Edition, John Wiley & Sons, 1979.

[3] F.M. Gardner, "Self-Noise in Synchronizers", IEEE Trans. on Comm., August 1980.

[4] Motorola, Inc. "FLEX^{TM} Protocol Specification and FLEX^{TM} Encoding and Decoding Requirments", 1996.

[5] F.M. Gardner, "Phase-locked Loops", John Wiley & Sons, 1979.