

VS1003 - MP3/WMA 오디오 코덱

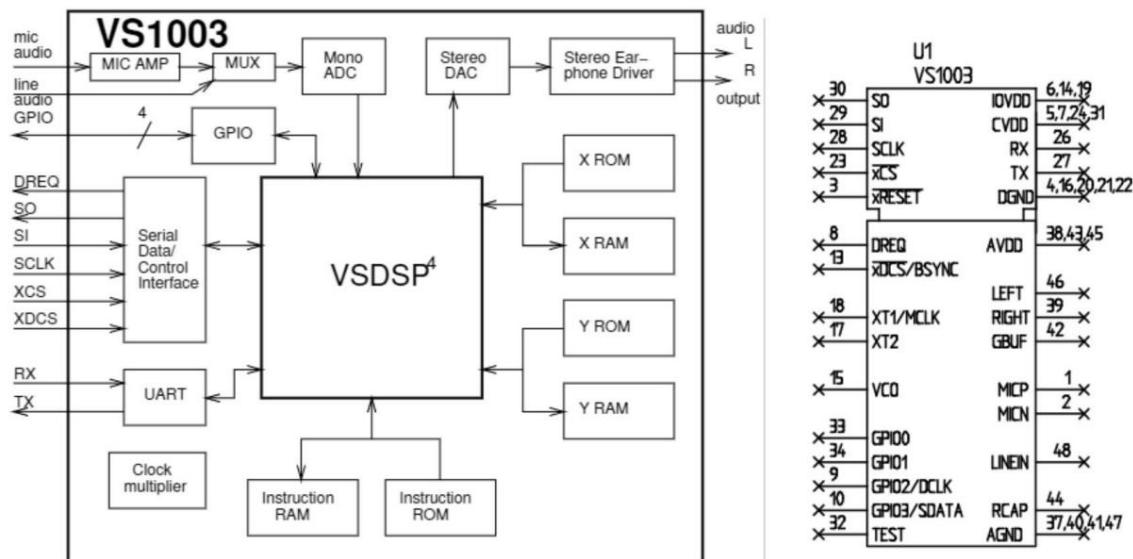
특징

- MPEG 1 & 2 오디오 레이어 III(CBR +VBR +ABR)을 디코딩합니다. WMA 4.0/4.1/7/8/9 모든 프로필(5-384kbit/s), WAV(PCM + IMA ADPCM); 일반 MIDI / SP-MIDI 파일
- マイ크 또는 라인 입력에서 IMA ADPCM 인코딩 • MP3 및 WAV에 대한 스트리밍 지원 • 저음 및 고음 컨트롤 • 단일 12~13MHz 클록으로 작동 • 내부 PLL 클록 체배기 • 저전력 작동 • 고품질 온침 스테레오 채널 간 위상 오차가 없는 DAC
- 30Q 부하를 구동할 수 있는 스테레오 이어폰 드라이버 • 아날로그용 별도의 작동 전압,

디지털 및 I/O • 사용자 코드용 5.5KiB 온칩 RAM / 데이터

- 직렬 제어 및 데이터 인터페이스 • 슬레이브 코 프로세서로 사용 가능 • 특수 애플리케이션을 위한 SPI 플래시 부트 • 디버깅 목적을 위한 UART • 새로운 기능은 소프트로 추가될 수 있습니다.

웨어 및 4개의 GPIO 핀



설명

VS1003은 단일 칩 MP3/WMA/MIDI 오디오 디코더 및 ADPCM 인코더입니다. 여기에는 고성능 독점 저전력 DSP 프로세서 코어 VS_DSP4, 작업 데이터 메모리, 사용자 애플리케이션용 5KiB 명령 RAM 및 0.5KiB 데이터 RAM, 직렬 제어 및 입력 데이터 인터페이스, 4개의 범용 I/O 핀, UART, 고품질 가변 샘플 속도 모노 ADC 및 스테레오 DAC, 이어 이어폰 증폭기 및 공통 버퍼.

VS1003은 직렬 입력 버스를 통해 입력 비트스트림을 수신하고 시스템 슬레이브로 수신합니다. 입력 스트림은 디코딩되어 디지털 볼륨 컨트롤을 통해 18비트 오버샘플링, 멀티비트, 시그마 델타 DAC로 전달됩니다. 디코딩은 직렬 제어 버스를 통해 제어됩니다. 기본 드 외에

코딩을 통해 DSP 효과와 같은 애플리케이션별 기능을 사용자 RAM 메모리에 추가할 수 있습니다.



실 VS1003

내용물

내용물

VS1003	1
목차	2
그림 목록	5
1 라이센스	6
2 면책조항	6
3 정의	6
4 특성 및 사양	7
4.1 절대 최대 정	
4.2 권장 작동 조건	7
4.3 아날로그 특성	7
4.4 소비전	8
4.5 디지털 특성	9
4.6 스위칭 특성 - 부트 초기화	9
4.7 일반적인 특성	9
4.7.1 10 라인 애날로그 ADC	10
4.7.2 4:7:3 RIGHT 및 1:4:7:3 LEFT 마이크 입력	10
4.7.3 BGA-49	11
5 패키지 및 핀 설명	12
5.1 LQFP-48	12
5.1.1 BGA-49	12
5.1.2 BGA-49 핀 설명	13
6 연결 다이어그램, LQFP-48	15
7 SPI 버스	16
7.1 일반 사항	16
7.2 SPI	16
7.2.1 VS10xx 기본 모드(새 모드)	16
7.2.2 VS1001 호환 모드	16
7.3 데이터 요청 핀 DREQ	17
7.4 직렬 데이터 인터페이스(SDI)용 직렬 프로토콜	17
7.5 SCI(Serial Command Interface)용 직렬 프로토콜	18
7.5.1 반	18
7.5.2 SCI 읽기	19
7.5.3 SCI 쓰기	19
7.6 SPI 타이밍 다이어그램	19
7.7 SM_SDINew 및 SM_SDIShareD 세트가 있는 SPI 예제	21
7.7.1 두 개의 SCI 쓰기	21



실 VS1003

내용물

7.7.2 2개의 SDI 바이트.....	21	7.7.3 2개의 SDI 바이트 중간에서 SCI 연산	22
8 기능 설명 23 8.1 주요 기능.....	23	8.2 지원되는 오디오 코덱	23 8.2.1 지원되는 MP3(MPEG
레이어 III) 형식	23	8.2.2 지원되는 WMA 형식	24 8.2.3 지원되는 RIFF WAV 형식
식	25	8.2.4 지원하는 ML미 형식	25
식	26	8.3 VS1003의 데이터 흐름	26 8.3 VS1003의 데이터 흐름
아이스(SDI).....	27	8.4 직렬 데이터 인터페이스.....	27 8.5초 직렬 제어 인터페이스.....
(SCI)터.....	28	8.6 SCI 레지스.....	28 8.6.1
SCI_MODE(RW)	29	8.6.2	
SCI_STATUS(RW)	31	8.6.3	
SCI_BASS(RW)	31	8.6.4	
SCI_CLOCKF(RW)	32	8.6.5	
SCI_DECODE_TIME(RW)	33	8.6.6	
SCI_AUDATA(RW).....	33	8.6.7 SCI_WRAM	
(RW).....	33	8.6.8	
SCI_WRAMADDR(W)	33	8.6.9 SCI_HDAT0 및	
SCI_HDAT1(R).....	34	8.6.10	
SCI_AIADDR(RW)	35	8.6.11	
SCI_VOL(RW)	36	8.6.12 SCI_AICTRL[x]	
(RW).....	36		
9 작동 37 9.1 클럭ing.....	37	9.2 하드웨어 리셋.....	37 9.2 하드웨어 리셋.....
셋.....셋.....	37	9.3 소프트웨어 리셋.....	37 9.4 ADPCM 녹화
화화	38	9.4 ADPCM 모드 활성화	38 ADPCM 모드 활성화
화	38	9.4.2 IMA ADPCM 데이터 읽기	38 9.4.2 IMA ADPCM 데이터 읽기
기	38	9.4.3 RIFF 헤더 추	
9.4.1 가.....	39	9.4.4 ADPCM 데이터 재생	39 9.4.4 ADPCM 데이터 재생
생	40	9.4.5 샘플 속도 고려 사항	40 9.4.5 샘플 속도 고려 사항
항	40	9.4.6 예제 코드	40 9.4.6 예제 코드
드통.....	40	9.5 SPI 부트	40 9.5 SPI 부트
42 9.6 재생/디코드	42		
9.7 PCM 데이터 공급.....	42	9.8 SDI 테스트	42 9.8 SDI 테스트
트	43	9.8.1 사인 테스트	43 9.8.1 사인 테스트
트.....	43		
9.8.2 핀 테스트.....	44		
9.8.3 메모리 테스트.....	44	9.8.4 SCI 테스트	44 9.8.4 SCI 테스트
트.....	44		
VS1003 레지스터 10개	45		
10.1 이 장을 읽어야 하는 사람	45	10.2 프로세서 코어	45 10.2 프로세서 코어
어	45		
10.3 VS1003 메모리 맵.....	45		



실 VS1003

내용물

10.4 SCI 레지스터.....	
45 10.5 직렬 데이터 레지스터.....	46
10.6 DAC 레지스터	47 10.7
GPIO 레지스터.....	47 10.8 인터럽트 레
지스터	48 10.9 A/D 변조기 레지스
터	49 10.10 워치
독	50 10.10.1 레지스
터.....	50 10.11 UART(Universal
Asynchronous Receiver/Transmitter)	51 10.11.1 레지스
터.....	51 10.11.2 상태
UARTx_STATUS	51 10.11.3 데이터
UARTx_DATA.....	52 10.11.4 데이터 하이
UARTx_DATAH.....	52 10.11.5 분배기
UARTx_DIV.....	52 10.11.6 인터럽트 및 연
산	53 10.12 타이
머.....	54
10.12.1 레지스터.....	
54 10.12.2 구성 TIMER_CONFIG	54 10.12.3
TIMER2ENABLE 구성	55 10.12.4 타이머 X 시작값
TIMER_Tx[L/H]	55 10.12.5 타이머 X 카운터
TIMER_TxCNT [L/H].....	55 10.12.6 인터럽
트	55 10.13 시스템
벡터 태그.....	56 10.13.1 AudioInt,
0x20.....	56 10.13.2 Scilnt,
0x21	56 10.13.3 DataInt,
0x22.....	56 10.13.4 모듈런트,
0x23.....	56 10.13.5 TxInt,
0x24	57 10.13.6 RxInt,
0x25	57 10.13.7 타이머,
0x26.....	57 10.13.8 타임린트,
0x27	57 10.13.9 사용자 코덱,
0x0.....	58 10.14 시스템 벡터 함
수.....	58 10.14.1 WriteRam(),
0x2.....	58 10.14.2 ReadRam(),
0x4.....	58 10.14.3 데이터바이트(),
0x6.....	59 10.14.4 GetDataByte(),
0x8.....	59 10.14.5 GetDataWords(), 옥
사	59 10.14.6 재부팅(),
0xc.....	59
11 최신 문서 버전 변경 사항	60
12 연락처 정보	61



그림 목록

1	LINEIN 핀의 측정된 ADC 성능.....	10 2 MIC 핀의 측정된 ADC 성능(차 동)	11 4
	10 3 RIGHT(또는 LEFT) 출력의 성능 측정...		
	RIGHT(또는 LEFT) Ijtput의 일반적인 스펙트럼	11	
5	핀 구성, LQFP-48.....	12 6핀 구성, BGA-49.....	
	15 8 BSYNC 신호 2비트 QEP 48을 사용한 일반적인 결선 송.....	18 9 BSYNC 신호 - 2바이트 전 송.....	
	18 10 SCI 단어 읽 기	19 11 SCI 워드 쓰 기.....	
	19 12 SPI 타이밍 다이어그 램	20 13 두 개의 SCI 작 업.....	
	21 14 2개의 SDI 바이 트.....	21 15 SCI 작업으로 분리된 두 개 의 SDI 바이트.....	
	22 16 VS1003의 데이터 흐 름.....	27 17 8kHz 샘플 속도의 ADPCM 주파수 응 답.....	
	30 18 사용자의 메모리 맵	46 19 RS232 직렬 인터페이스 프로토 콜.....	
	51		

1 라이센스

VS1003에는 Microsoft의 WMA 디코딩 기술이 포함되어 있습니다.

이 제품은 Microsoft의 특정 지적 재산권에 의해 보호되며 Microsoft의 라이선스 없이 사용하거나 추가로 배포할 수 없습니다.

2 면책조항

모든 속성과 수치는 변경될 수 있습니다.

3 정의

ABR 평균 비트레이트. 스트림의 비트 전송률은 지역에 따라 다를 수 있지만 주어진 숫자에 가깝게 유지됩니다.

더 긴 시간 동안 평균했을 때.

B 바이트, 8비트.

b 비트.

CBR 고정 비트레이트. 스트림의 비트 전송률은 각 압축 블록에 대해 동일합니다.

CBUF 헤드폰 공통 버퍼. DC 전압을 출력합니다.

GBUF CBUF 와 동일.

Ki "Kibi" = $2 \cdot 10 = 1024$ (IEC 60027-2).

Mi "Mebi" = $2^{20} = 1048576$ (IEC 60027-2).

VS1003 제어용 SPI 버스인 SCI 직렬 제어 인터페이스.

VS1003 비트스트림 데이터용 SPI 버스인 SDI 직렬 데이터 인터페이스.

VBR 가변 비트레이트. 비트 전송률은 원본 자료의 복잡성에 따라 달라집니다.

VS_DSP VLSI 솔루션의 DSP 코어.

VSIDE VLSI 솔루션의 통합 개발 환경입니다.

여 말씀. VS_DSP에서 명령어는 32비트이고 데이터어는 16비트입니다.

4 특성 및 사양

4.1 절대 최대 정격

모수	최소 기호		최대	단위
아날로그 포지티브 공급	AVDD -0.3	CVDD	2.85	에
디지털 포지티브 공급	-0.3		2.85	에
I/O 포지티브 서플라이	IOVDD -0.3		3.6	에
모든 디지털 출력의 전류			± 50	엠마
모든 디지털 입력의 전압			-0.3 IOVDD+ 0.3 1V	
작동 온도		-40	+85	°C
보관 온도		-65	+150	°C

1 3.6V를 초과하지 않아야 함

4.2 권장 작동 조건

모수	상징	분	유형	최대 단위
주변 작동 온도		-40		+85 °C
아날로그 및 디지털 접지 1	AGND DGND		0.0	에
포지티브 아날로그	학과	2.6	2.8 2.85V 2.5 2.85V	2.8
포지티브 디지털	CVDD	2.4	3.6V 12.288MHz 3.6V 12.288MHz	3.6V 12.288MHz
입출력 전압	IOVDD	CVD-0.6V	1MHz 4.5x 4	
입력 클록 주파수 1 2 3	엑스탈리	12		
내부 클럭 주파수	씨엘키	12		
내부 클록 승수			액체 산소 3.0배	
마스터 클록 둑티 사이클		40	50	60%

1 래치업 내성을 위해 가능한 한 장치에 가깝게 함께 연결해야 합니다.

2 올바른 속도로 재생할 수 있는 최대 샘플 속도는 XTAL/256입니다.

따라서 올바른 속도로 48kHz를 재생하려면 X7ALI가 12.288MHz 이상이어야 합니다.

3 재설정 값은 I.Ox입니다. 권장 SC_MULT=3.0x, SC_ADD=1.0x(SCI_CLOCKF=0x9000).

4 52.0MHz는 전체 CVD 범위에 대한 최대 클록입니다.

(4.0 x 12.288MHz=49.152MHz 또는 4.0 x 13.0MHz=52.0MHz)

4.3 아날로그 특성

달리 명시되지 않는 한: AVDD=2.85V, CVDD=2.5V, IOVDD=-2.8V, TA=-25..+70°C, X7ALI=12.288MHz, DAC는 1307.894Hz 전체 범위 출력 사인파로 테스트됨, 측정 대역폭 20 ..20000Hz, 아날로그 출력 부하: LEFT에서 GBUF 30Q로, RIGHT에서 GBUF 30Q로. 마이크 테스트 진폭 50mVpp, f=1kHz, 라인 입력 테스트 진폭 2.2Vpp, f=1kHz.

모수	기호	최소	유형	최대	단위	18	0.1	>90	SNR	70	s	83	d	50
DAC 해상도				75	40	-0.5	± 0.2	-0.1	비트					
총 고조파 왜곡	THD					0.3%	dB							
다이내믹 레인지(DAC 음소거 해제, A-기증치)	루피아													
S/N 비율(풀 스케일 신호)														
채널 간 격리(혼선)														데시벨
GBUF를 사용한 채널 간 격리(Cross Talk)														데시벨
채널 간 게인 불일치						0.5dB	0.1							
주파수 응답						1.7Vpp	5							데시벨
풀 스케일 출력 전압(Peak-to-peak)		1.3		1.5	1									
선형 위상의 편차														정밀
아날로그 출력 부하 저항	AOL	16	30	2										
아날로그 출력 부하 용량						큐	100pF							
마이크 입력 증폭기 게인	MICG			26										데시벨
마이크 입력 진폭				50		140	3			mVpp	AC			
마이크 총 고조파 왜곡	MTHD			0.02		0.10%								
마이크 S/N 비율	MSNR	5	("		68						dB			
라인 입력 진폭					2200	2800	3			mVpp	AC			
라인 입력 총 고조파 왜곡	LTHD			0.015	0.10%									
라인 입력 S/N 비율	LSNR	60	5		86						dB			
라인 및 마이크 입력 임피던스					100					kQ				

일반적인 값은 Lot 4234011, Week Code 0452의 약 5000개 장치에서 측정됩니다. 1 3.0V는 모노 차이 사운드의 +to- + 배선으로 달성할 수 있습니다.

2 AOLR은 훨씬 낮을 수 있지만 일반 왜곡 성능보다 낮을 수 있습니다.

3 일반적인 진폭 이상에서는 고조파 왜곡이 증가합니다.

⁴ 비기증, A-기증은 약 3dB 더 좋습니다.

5 생산 테스터의 소음 수준으로 인해 한계가 낮습니다.

4.4 소비 전력

MPEG 1.0 Layer-3 128kbit/s 샘플 및 생성된 사인으로 테스트되었습니다. 전체 볼륨으로 출력합니다.

X7ALI 12.288MHz. 내부 클록 승수 3.0x. CVDD = 2.5V, AVDD = 2.8V.

모수	최소	유형	최대	단위
전원 소비 AVDD, 리셋	0.6		5.0	// t
전원 소비 전력 CVD, 리셋, +25°C	3.7		40.0	//A
전원 소비 전력 CVD, 리셋, +85°C			200.0	//A
전원 소비량 AVDD, 사인 테스트, 30Q + GBUF	36.9			엄마
전원 소비량 CVDD, 사인 테스트	12.4			엄마
전원 소비 AVDD, 무부하	7.0			엄마
전원 소비 AVDD, 출력 부하 30Q	10.9			엄마
전원 공급 장치 소비 AVDD, 30Q + GBUF	16.1			엄마
전원 공급 장치 소비 CVDD	17.5			엄마

4.5 디지털 특성

모수	상정	분	유형	최대	단위
높은 수준의 입력 전압 낮은 수준의		0.7xIOVDD		IOVDD+ 0.3	1V
입력 전압 높은 수준의 출력 전압(Io		-0.2		0.3xIOVDD	V
= -1.0mA) 낮은 수준의 출력 전압(Io = 1.0mA) 입력 누설 전		0.7xIOVDD			에
류 SPI 입력 클록 주파수 모든 출력 핀의 상승 시간, 부하 =				0.3xIOVDD	V
50pF		-1.0		1.0	// t
2				-CLK1- 7	메가헤르츠
				50	NS

1 3.6V를 초과하면 안 됩니다.

2 SCI 읽기 값입니다. SCI 및 SDI 쓰기는 CL^ i 허용

4.6 스위칭 특성 - 부트 초기화

모수	최소	기호	최대	단위
XRESET 활성 시간	2			엑스탈리
소프트웨어 준비에 대한 XRESET 비활성	16600	50000	1	엑스탈리
파워 온 리셋, CVD까지의 상승 시간	10			대

1 DREQ는 초기화가 완료되면 상승합니다. 그 전에는 데이터나 명령을 보내면 안 됩니다.

4.7 일반적인 특성

4.7.1 Line input ADC

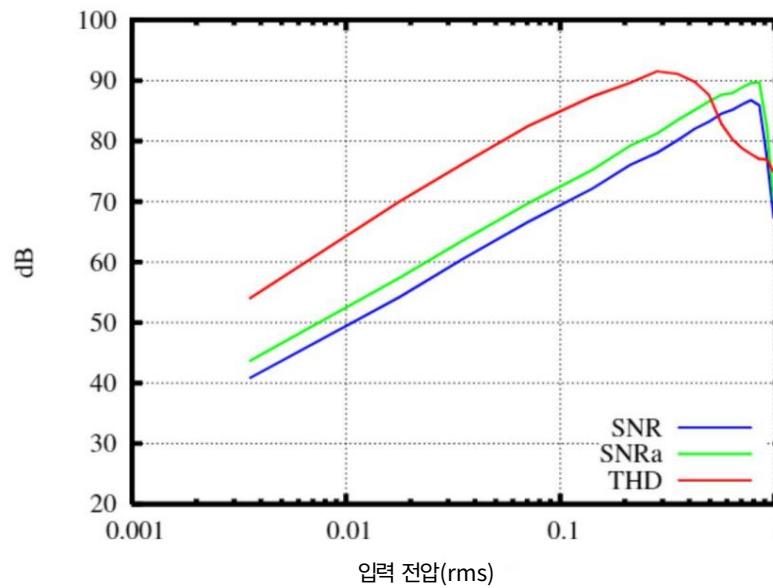


그림 1: LINEIN 핀의 측정된 ADC 성능.

LINEIN 핀의 측정된 ADC 성능. X축은 1kHz 사인 입력의 rms 진폭입니다.

곡선은 비가중 신호 대 잡음비(파란색), A 가중치 신호 대 잡음비(녹색) 및 비가중 신호 대 왜곡비(빨간색)입니다. ADC의 샘플링 속도는 48kHz(마스터 클럭 12.288MHz)이고 잡음은 0~20kHz에서 계산됩니다.

4.7.2 마이크 입력 ADC

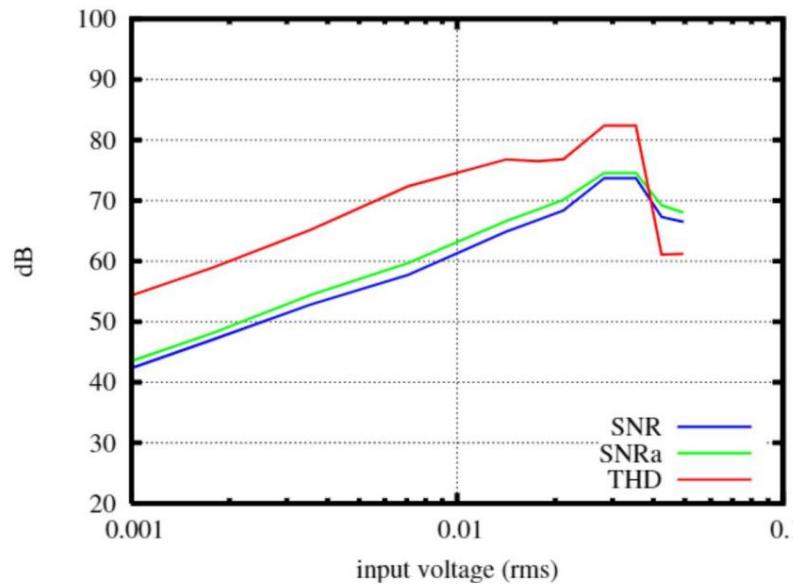


그림 2: MIC 핀의 측정된 ADC 성능(차동).

MIC 핀의 측정된 ADC 성능(차동). 기타 설정은 그림 1과 동일합니다.

4.7.3 오른쪽 및 왼쪽 출력

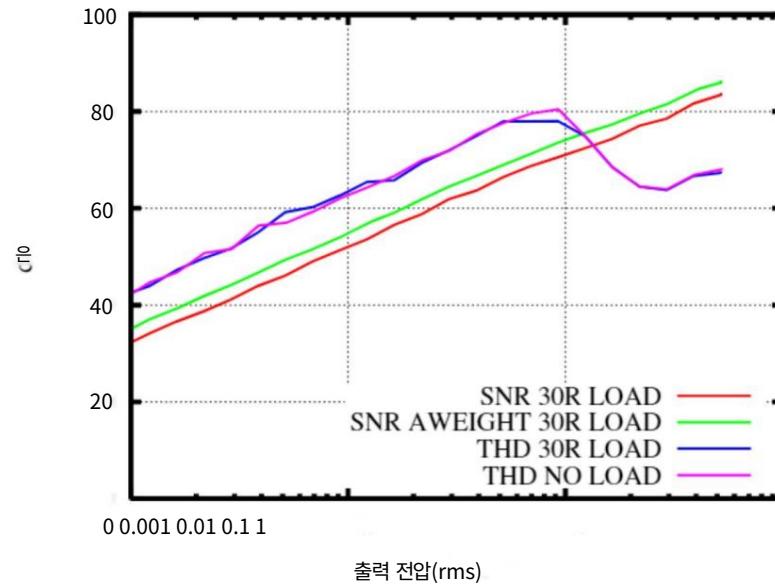


그림 3: RIGHT(또는 LEFT) 출력의 측정된 성능.

1kHz 생성 사인으로 RIGHT(또는 LEFT) 출력의 측정된 성능입니다. DAC의 샘플링 속도는 48kHz(마스터 클럭 12.288MHz)이며 잡음은 0~20kHz에서 계산됩니다.

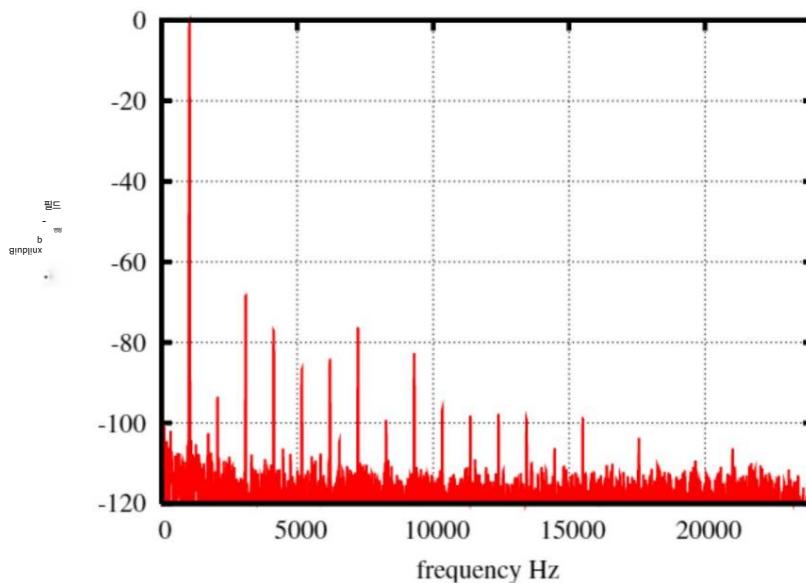


그림 4: RIGHT(또는 LEFT) 출력의 일반적인 스펙트럼.

최대 레벨 및 30옴 부하가 있는 RIGHT(또는 LEFT) 출력의 일반적인 스펙트럼. 설정은 그림 3과 동일합니다.

5 패키지 및 핀 설명

5.1 패키지

LPQFP-48 및 BGA-49는 모두 납(Pb)이 없고 RoHS 준수 패키지입니다. RoHS는 특정 유해 물질의 사용 제한에 관한 Directive 2002/95/EC의 짧은 이름입니다.

전기 및 전자 장비의 물질.

5.1.1 LQFP-48

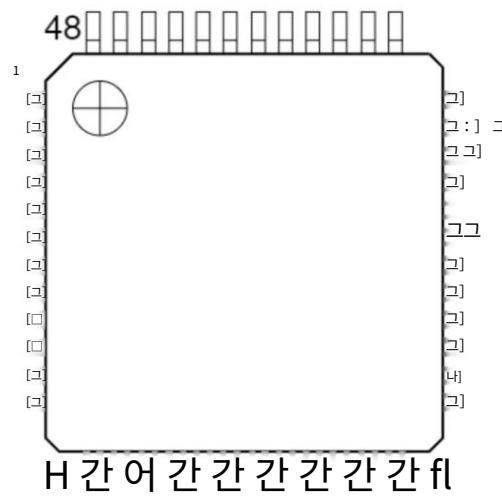


그림 5: 핀 구성, LQFP-48.

LQFP-48 패키지 치수는 [http://www.vlsi.fi/에 있습니다.](http://www.vlsi.fi/)

5.1.2 BGA-49

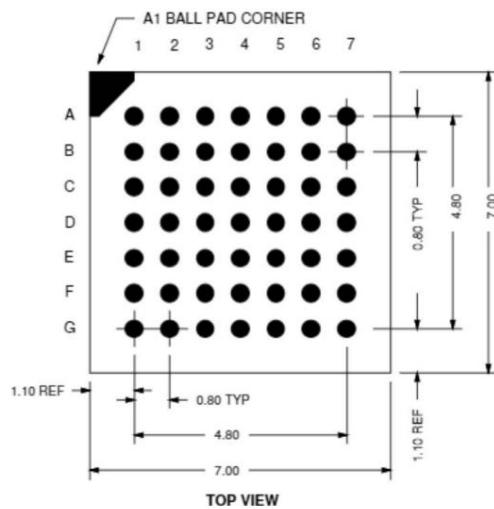


그림 6: 핀 구성, BGA-49.

BGA-49 패키지 치수는 [http://www.vlsi.fi/에 있습니다.](http://www.vlsi.fi/)

5.2 LQFP-48 및 BGA-49 핀 설명

핀 이름	LQFP 48 핀	BGA49 공	핀 유형	기능
MICP	1	C3	알	포지티브 차동 마이크 입력, 자체 바이어스
미씨엔	2	C2	알	네거티브 차동 마이크 입력, 자체 바이어스
XRESET	3	B1	에서	액티브 로우 비동기 리셋
DGNDO	4	D2	DGND 코어 및 I/O 접지	
CVDDO	5	C1	CPWR 코어 전원 공급 장치	
IOVDDO	6	D3	IOPWR I/O 전원 공급 장치	
CVDD1	7	D1	CPWR 코어 전원 공급 장치	
도대체	8	E2	-하다	데이터 요청, 입력 버스
GPIO2 / DCLK1	9	E1	부분	범용 IO 2/직렬 입력 데이터 버스 클록
GPIO3 / SDATA1 10		F2	미 O 범용 IO 3 / 직렬 데이터 입력	
XDCS/BSYNC1 13		E3	에서	데이터 칩 선택/바이트 동기화
IOVDD1 14		F3	IOPWR I/O 전원 공급 장치	
VCO	15	G2	-하다	테스트 전용(클럭 VCO 출력)
DGND1	16	F4	DGND 코어 및 I/O 접지	
엑스탈로	17	G3	에게	크리스탈 출력
엑스탈리	18	E4	알	크리스탈 입력
IOVDD2	19	G4	IOPWR I/O 전원 공급 장치	
IOVDD3		F5	IOPWR I/O 전원 공급 장치	
DGND2	20		DGND 코어 및 I/O 접지	
DGND3	21	G5	DGND 코어 및 I/O 접지	
DGND4	22	F6	DGND 코어 및 I/O 접지	
XCS	23	G6	에서	칩 선택 입력(액티브 로우)
CVDD2	24	G7	CPWR 코어 전원 공급 장치	
RX	26	E6	에서	UART 수신, 사용하지 않을 경우 IOVDD에 연결
텍사스	27	F7	-하다	UART 전송
SCLK	28	D6	에서	직렬 버스용 클록
그리고	29	E7	에서	직렬 입력
그래서	30	D5	DO3	직렬 출력
CVDD3	31	D7	CPWR 코어 전원 공급 장치	
테스트 32		C6	에서	테스트용으로 예약됨, IOVDD에 연결
GPIO0/스파이부트 33		C7	미 O 범용 IO 0 / SPI BOOT, 100kQ 풀다운 저항 사용 1 2	
GPIO1	34	B6	미 O 범용 IO	
아그도	37	C5	APWR 아날로그 접지, 저잡음 기준	
AVDDO	38	B5	APWR 아날로그 전원 공급 장치	
오른쪽	39	A6	에게	오른쪽 채널 출력
AGND1	40	B4	APWR 아날로그 접지	
AGND2	41	A5	APWR 아날로그 접지	
GBUF	42	C4	에게	헤드폰용 공통 버퍼
AVDD1	43	A4	APWR 아날로그 전원 공급 장치	
RCAP	44	B3	AIO	참조용 필터링 커패시턴스
AVDD2	45	A3	APWR 아날로그 전원 공급 장치	
왼쪽	46	B2	에게	왼쪽 채널 출력
AGND3	47	A2	APWR 아날로그 접지	
리네인	48	A1	알	라인 입력

1 첫 번째 핀 기능은 새 모드에서 활성화되고 후자는 호환성 모드에서 활성화됩니다.

2 풀다운 저항을 사용하지 않는 한 SPI Boot를 시도합니다. 자세한 내용은 9.5장을 참조하십시오.



VS1003

5 패키지 및 PIN 설명

핀 유형:

유형 설명	
I/O	디지털 입력, CMOS 입력 패드
DO	디지털 출력, CMOS 입력 패드
DIO	디지털 입/출력
DO3	디지털 출력, CMOS Tri-stated 출력 인수
AI	아날로그 입력

유형	설명
에게	아날로그 출력
AIO	아날로그 입력/출력
APWR	아날로그 전원 핀
DGND 코어	또는 I/O 접지 핀
CPWR 코어	전원 핀
IOPWR I/O	전원 핀

BGA-49에서 연결되지 않은 공은 A7, B7, D4, E5, F1, G1입니다.

LQFP-48에서 비연결 핀은 11, 12, 25, 35, 36입니다.

6 연결 다이어그램, LQFP-48

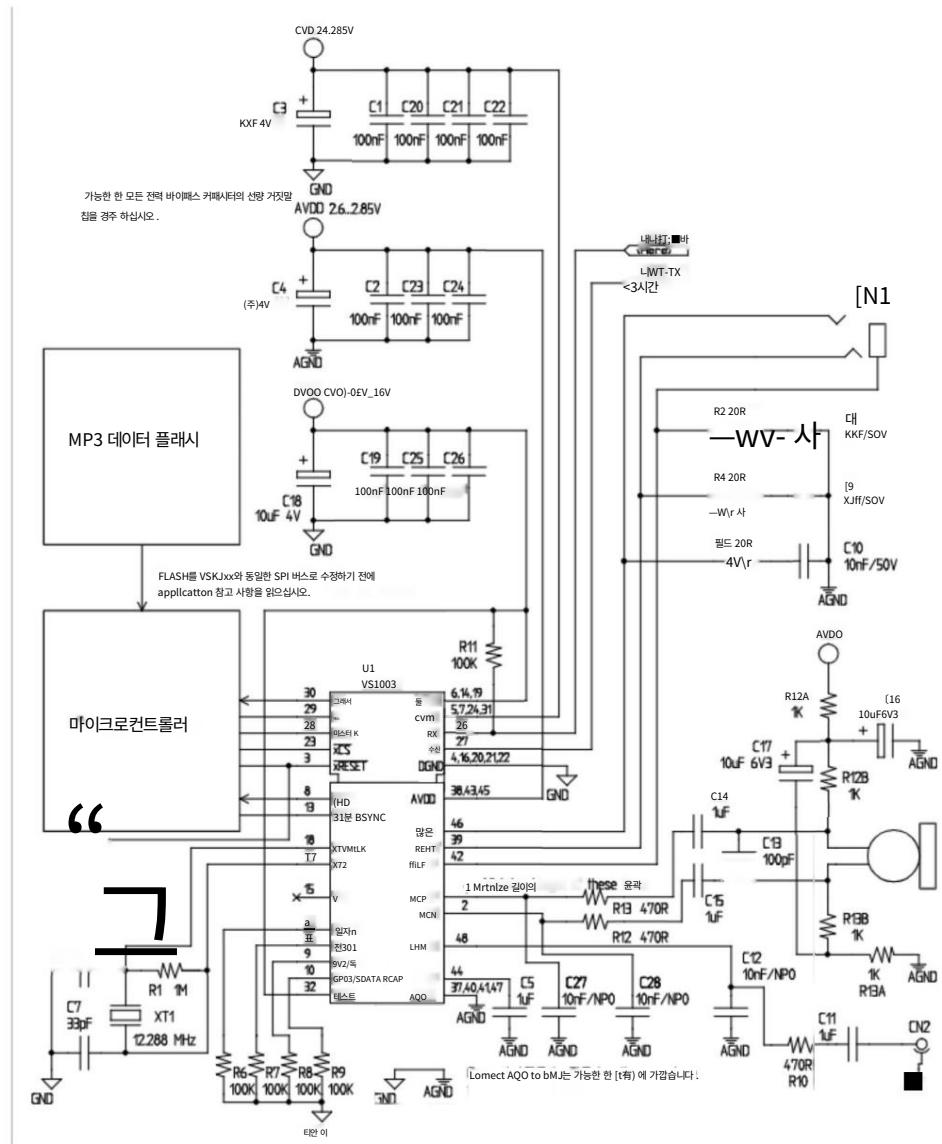


그림 7: LQFP-48을 사용한 일반적인 연결 다이어그램.

공통 버퍼 GBUF는 이어폰의 공통 전압(1.24V)에 사용할 수 있습니다. 이렇게 하면 라인 출력에 큰 절연 커패시터가 필요하지 않으므로 VS1003의 오디오 출력 핀을 이어폰 커넥터에 직접 연결할 수 있습니다.

GBUF는 어떤 경우에도 접지에 연결해서는 안 됩니다. GBUF를 사용하지 않는 경우 LEFT 및 RIGHT에 커플링 커패시터를 제공해야 합니다. GBUF를 안정적으로 유지하려면 GBUF를 사용하지 않는 경우에도 저항과 커패시터를 항상 준비해야 합니다. 자세한 내용은 애플리케이션 노트를 참조하십시오.

사용하지 않는 GPIO 핀에는 풀다운 저항이 있어야 합니다.

UART를 사용하지 않는 경우 RX는 IOVDD에 연결하고 TX는 연결을 해제해야 합니다.

X7AL0에 외부 부하를 연결하지 마십시오.

참고: 이 연결은 SM_SDINew가 활성 상태라고 가정합니다(8.6.1장 참조). SM_SDISHARE도 사용되는 경우 xDCS는 높게 연결되어야 합니다(챕터 7.2.1 참조).

7 SPI 버스

7.1 일반

일부 Motorola 장치에서 원래 사용되었던 SPI 버스는 VS1003의 직렬 데이터 인터페이스 SDI(7.4 및 8.4장) 및 직렬 제어 인터페이스 SCI(7.5 및 8.5장) 모두에 사용되었습니다.

7.2 SPI 버스 핀 설명

7.2.1 VSIOxx 기본 모드(새 모드)

이러한 모드는 SM_SDINEW가 1(시작 시 기본값)로 설정될 때 VS1003에서 활성화됩니다. DCLK 및 SDATA는 데이터 전송에 사용되지 않으며 범용 I/O 핀(GPIO2 및 GPIO3)으로 사용할 수 있습니다. BSYNC 기능이 데이터 인터페이스 칩 선택(XDCS)으로 변경됩니다.

SDI 핀 SCI 핀 설명	
XDCS XCS	액티브 로우 칩 선택 입력. 높은 수준은 직렬 인터페이스를 대기 모드, 현재 작업 종료. 높은 수준은 또한 직렬 출력(SO)을 높은 임피던스 상태로 만듭니다. SM_SDISHARE가 10이면 XDCS 핀은 사용하지 않지만 XCS를 반전시켜 내부적으로 신호를 발생시킨다.
SCK	직렬 클럭 입력. 직렬 클럭은 내부적으로 레지스터 인터페이스의 마스터 클럭으로도 사용됩니다. SCK는 게이트형 또는 연속형일 수 있습니다. 두 경우 모두 XCS가 낮아진 후 첫 번째 상승 클록 에지가 기록될 첫 번째 비트를 표시합니다.
그리고	직렬 입력. 칩 선택이 활성화되면 상승 CLK 에지에서 SI가 샘플링됩니다.
• SO	직렬 출력. 읽기에서 데이터는 하강 SCK 에지에서 이동됩니다. 쓰기에서 SO는 높은 임피던스 상태에 있습니다.

7.2.2 VS1001 호환 모드

이 모드는 SM_SDINEW가 0으로 설정될 때 활성화됩니다. 이 모드에서 DCLK, SDATA 및 BSYNC가 활성화됩니다.

SDI 핀 SCI 핀 설명	
-	XCS 액티브 로우 칩 선택 입력. 높은 수준은 직렬 인터페이스를 대기 모드, 현재 작업 종료. 높은 수준은 또한 직렬 출력(SO)을 높은 임피던스 상태로 만듭니다.
비싱크	- SDI 데이터는 BSYNC의 상승 에지와 동기화됩니다.
DCLK SCK	직렬 클록 입력. 직렬 클럭은 내부적으로 마스터로도 사용됩니다. 레지스터 인터페이스를 위한 클럭. SCK는 게이트형 또는 연속형일 수 있습니다. 두 경우 모두 XCS가 낮아진 후 첫 번째 상승 클록 에지가 기록될 첫 번째 비트를 표시합니다.
SDATA	그리고 직렬 입력. SI는 XCS가 낮을 경우 상승 SCK 에지에서 샘플링됩니다.
-	SO 직렬 출력. 읽기에서 데이터는 하강 SCK 에지에서 이동됩니다. 쓰기에서 SO는 높은 임피던스 상태에 있습니다.

7.3 데이터 요청 핀 DAMN

DREQ 핀/신호는 VS1003의 FIFO가 데이터를 수신할 수 있는지 여부를 알리는 데 사용됩니다. DREQ가 높으면 VS1003은 최소 32바이트의 SDI 데이터 또는 하나의 SCI 명령을 사용할 수 있습니다. 이러한 기준이 충족되지 않으면 DREQ가 낮아지고 발신자는 새 데이터 전송을 중지해야 합니다.

32바이트 안전 영역으로 인해 송신자는 DREQ의 상태를 확인하지 않고 한 번에 최대 32바이트의 SDI 데이터를 보낼 수 있으므로 저속 마이크로 컨트롤러에서 VS1003을 보다 쉽게 제어할 수 있습니다.

참고: DREQ는 바이트 전송 중에도 언제든지 낮거나 높을 수 있습니다. 따라서 DREQ는 더 많은 바이트를 보낼지 여부를 결정하는 데에만 사용해야 합니다. 이미 시작된 전송을 중단해서는 안 됩니다.

참고: VS1002까지의 VS10XX 제품에서 DREQ는 SDI에만 사용되었습니다. VS1003에서 DREQ는 SCI의 상태를 알리는 데에도 사용됩니다.

DREQ가 낮을 때 여전히 SCI 명령을 보내려는 경우가 있습니다. DREQ는 SDI와 SCI 간에 공유되므로 SDI가 수신할 준비가 되지 않은 경우 SCI 명령이 실행되었는지 확인할 수 없습니다. 이 경우 SCI 명령이 누락되지 않았는지 확인하기 위해 모든 SCI 명령 후에 충분히 긴 자연 시간이 필요합니다. 섹션 8.6의 SCI 레지스터 표는 각 SCI 레지스터 쓰기에 대한 최악의 처리 시간을 제공합니다.

7.4 직렬 데이터 인터페이스(SDI)용 직렬 프로토콜

7.4.1 일반

직렬 데이터 인터페이스는 슬레이브 모드에서 작동하므로 외부 회로에서 DCLK 신호를 생성해야 합니다.

데이터(SDATA 신호)는 DCLK의 상승 또는 하강 에지에서 클록 인될 수 있습니다(8.6장).

VS1003은 데이터 입력이 바이트 동기화된 것으로 가정합니다. SDI 바이트는 SCI_MODE의 내용에 따라 MSb 또는 LSb 먼저 전송될 수 있습니다(8.6.1장).

펌웨어는 SDI가 지원하는 최대 비트 전송률을 수용할 수 있습니다.

7.4.2 VSIOxx 기본 모드의 SDI(새 모드, 권장)

VSIOxx 기본 모드(SM_NEWMODE가 1)에서 바이트 동기화는 XDCS에 의해 이루어집니다.

데이터 바이트 전송이 진행되는 동안 XDCS의 상태는 변경되지 않을 수 있습니다. VS1003을 사용하는 보드에 결함이 있더라도 항상 데이터 동기화를 유지하려면 때때로 XDCS를 켜는 것이 좋습니다. VS1003이 동기화되었습니다.

SM_SDISHARE가 1이면 XCS 입력을 반전하여 XDCS 신호가 내부적으로 생성됩니다.

새 설계의 경우 VSIOxx 기본 모드를 사용하는 것이 좋습니다.

7.4.3 VS1001 호환 모드의 SDI

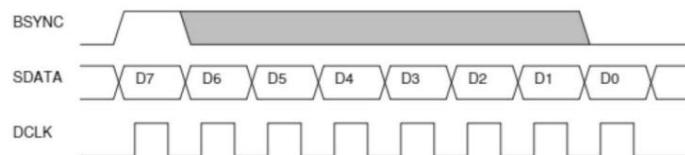


그림 8: BSYNC 신호 - 1바이트 전송.

VS1003이 VS1001 호환 모드에서 실행 중이면 입력 비트스트림의 올바른 비트 정렬을 보장하기 위해 BSYNC 신호가 생성되어야 합니다. 첫 번째 DCLK 샘플링 에지(선택한 극성에 따라 상승 또는 하강)는 BSYNC가 하이인 동안 바이트의 첫 번째 비트(LSB, LSB-1차가 사용되는 경우, MSB, MSB-1차가 사용되는 경우)를 표시합니다.). 마지막 비트가 수신될 때 BSYNC가 T0이면 수신기는 활성 상태를 유지하고 다음 8비트도 수신됩니다.

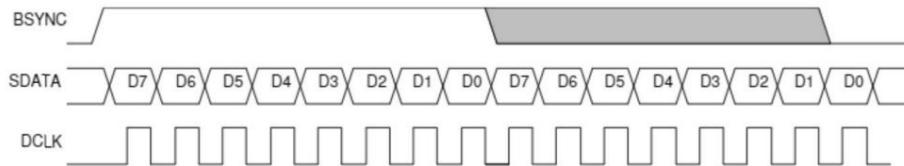


그림 9: BSYNC 신호 - 2바이트 전송.

7.4.4 패시브 SDI 모드

SM_NEWMODE가 0이고 SM_SDISHARE가 1이면 VS1001 호환 모드와 같지만 BSYNC 신호가 '1'일 때만 비트가 수신됩니다. BSYNC의 상승 에지는 여전히 동기화에 사용됩니다.

7.5 직렬 명령 인터페이스(SCI)용 직렬 프로토콜

7.5.1 일반

직렬 명령 인터페이스 SCI(8.5장)의 직렬 버스 프로토콜은 명령 바이트, 주소 바이트 및 하나의 16비트 데이터 워드로 구성됩니다. 각 읽기 또는 쓰기 작업은 단일 레지스터를 읽거나 쓸 수 있습니다. 데이터 비트는 상승 에지에서 읽으므로 사용자는 하강 에지에서 데이터를 업데이트해야 합니다. 바이트는 항상 MSb를 먼저 보냅니다. XCS는 작업의 전체 기간 동안 낮아야 하지만 필요한 경우 비트 사이에 일시 중지를 가질 수 있습니다.

연산은 8비트 명령어 opcode로 지정됩니다. 지원되는 명령은 읽기 및 쓰기입니다. 아래 표를 참조하십시오.

지침		
이름	옵코드	작업
READ	Ob0000 0011	데이터 읽기
WRITE	Ob0000 0010	데이터 쓰기

참고: VS1003은 각 SCI 작업 후 DREQ를 낮게 설정합니다. 기간은 작업에 따라 다릅니다. DREQ가 다시 높기 전에는 새로운 SCI/SDI 작업을 시작할 수 없습니다.

7.5.2 SCI Read

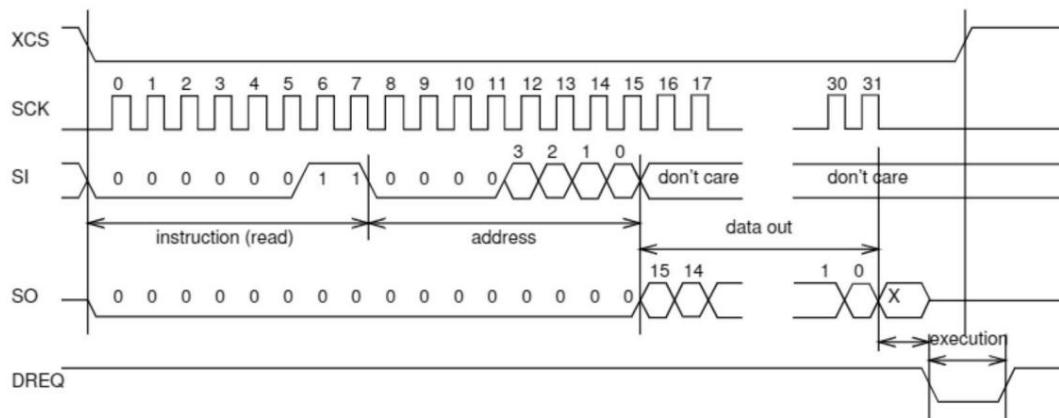


Figure 10: SCI Word Read

VS1003 레지스터는 그림 10과 같이 다음 순서를 사용하여 읽습니다. 먼저 XCS 라인을 로우로 당겨 장치를 선택합니다. 그런 다음 READ opcode(0x3)는 8비트 워드 주소가 뒤따르는 SI 라인을 통해 전송됩니다. 주소를 읽은 후 SI의 추가 데이터는 칩에서 무시됩니다. 수신된 주소에 해당하는 16비트 데이터는 SO 라인으로 이동됩니다.

XCS는 데이터가 이동된 후에 높게 구동되어야 합니다.

DREQ는 칩이 읽기 작업을 할 때 잠시 동안 로우로 구동됩니다. 이것은 매우 짧은 시간이며 특별한 사용자 주의가 필요하지 않습니다.

7.5.3 SCI 쓰기

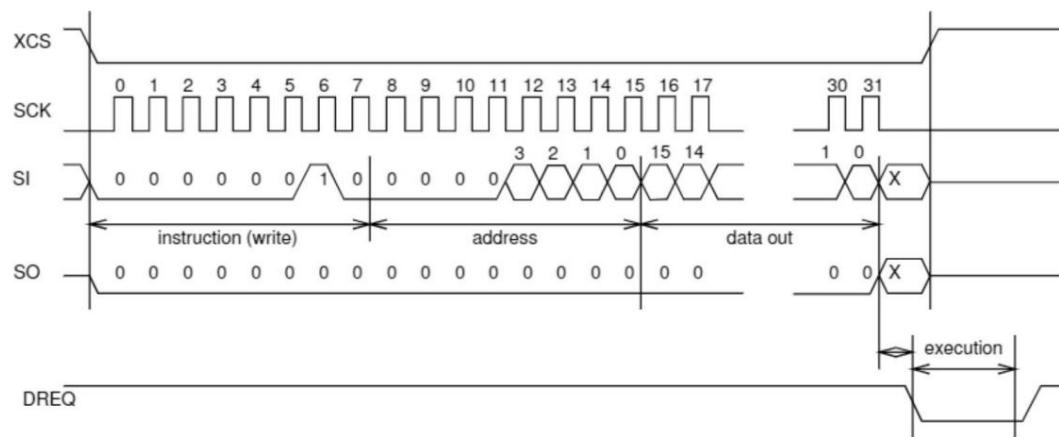


그림 11: SCI 단어 쓰기

VS1003 레지스터는 그림 11과 같이 다음 순서를 사용하여 기록됩니다. 먼저 XCS 라인을 로우로 당겨 장치를 선택합니다. 그런 다음 WRITE opcode(0x2)가 8비트 워드 주소가 뒤따르는 SI 라인을 통해 전송됩니다.

워드가 이동되고 마지막 클럭이 전송된 후 XCS를 하이로 끌어올려 WRITE 시퀀스를 종료해야 합니다.

마지막 비트가 전송된 후 DREQ는 그림에서 "실행"으로 표시된 레지스터 업데이트 기간 동안 로우로 구동됩니다. 시간은 레지스터와 그 내용에 따라 다릅니다(자세한 내용은 8.6장의 표 참조). 최대 시간이 마이크로컨트롤러에서 다음 SCI 명령 또는 SDI 바이트를 공급하는 데 걸리는 시간 보다 길면 DREQ가 다시 상승하기 전에 새로운 SCI/SDI 작업을 완료할 수 없습니다.

7.6 SPI 타이밍 다이어그램

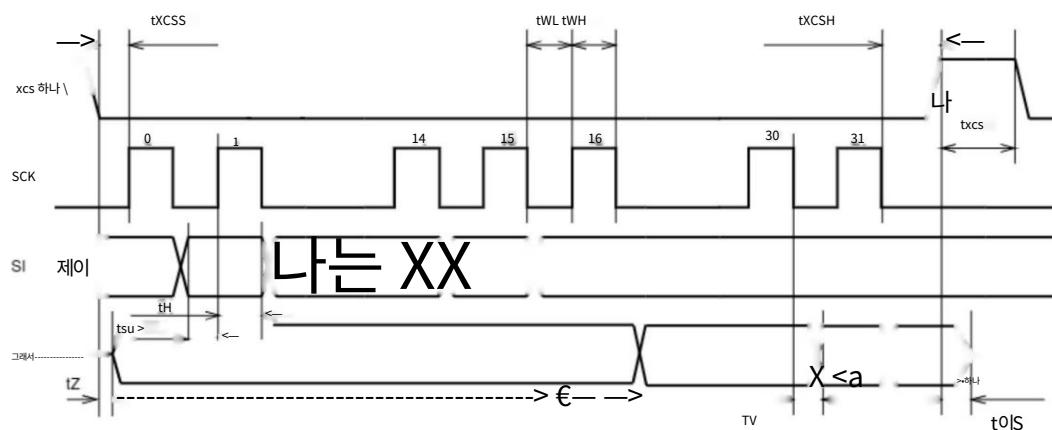


그림 12: SPI 타이밍 다이어그램.

기호 txcss	최소	최대	단위
tsu tH tz	5		NS
	0		NS
	2		CLKI 주기
	0		NS
tWL	2		CLKI 주기
tWH	2		CLKI 주기
TV	2(+25ns 1)	1	CLKI 주기
tXCSH			씨엘키
txcs	2		CLKI 주기 10ns
tDIS			

1 25ns는 핀에 10OpF 커패시턴스가 로드된 경우입니다. 커패시턴스가 낮을수록 시간이 더 짧습니다.

참고: tWL 및 tWH와 tH는 최소 2클럭 사이클이 필요하므로 비동기 클록과 함께 쉽게 사용할 수 있는 SPI 버스의 최대 속도는 VS1003의 내부 클록 속도 CLKI의 1/7입니다.

참고: 타이밍은 내부 클록 CLKI에서 파생되지만 시스템은 항상 I.Ox 모드에서 시작하므로 CLKI=X1ALI입니다.

7.7 SM_SDINew 및 SM_SDI_SHARED가 설정된 SPI 예제

7.7.1 Two SCI Writes

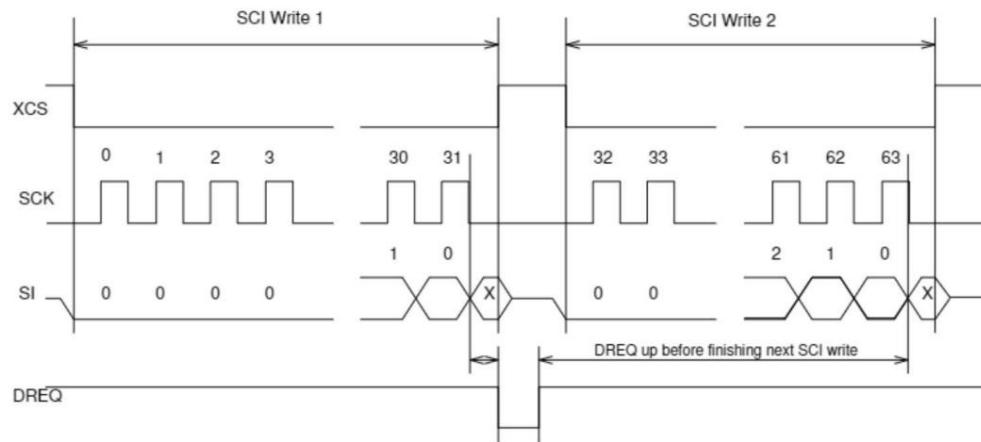


Figure 13: Two SCI Operations.

그림 13은 두 개의 연속적인 SCI 작업을 보여줍니다. xCS는 쓰기 사이에 비활성 상태로 올려야 합니다. 또한 그림과 같이 DREQ도 준수해야 합니다.

7.7.2 2개의 SDI 바이트

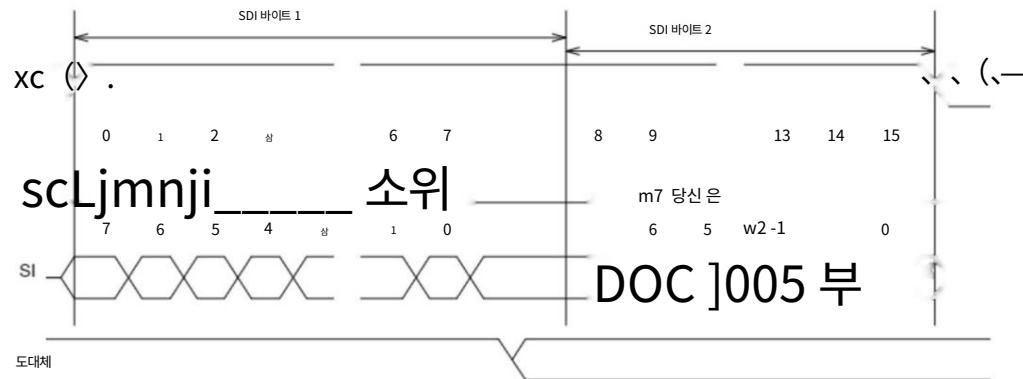


그림 14: 2개의 SDI 바이트.

SDI 데이터는 그림 14와 같이 xCS의 상승 에지로 동기화됩니다. 그러나 모든 바이트는 별도의 동기화가 필요하지 않습니다.

7.7.3 2개의 SDI 바이트 중간에서 SCI 동작

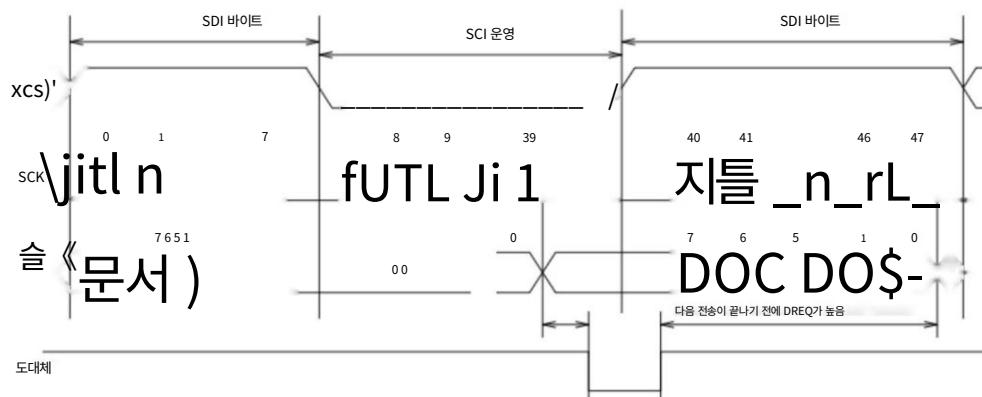


그림 15: "SCI 연산으로 분리된 2개의 SDI 바이트."

그림 15는 SDI 작업 사이에 SCI 작업이 포함되는 방법을 보여줍니다. xCS 에지는 SDI와 SCI를 모두 동기화하는 데 사용됩니다. 그림과 같이 DREQ를 존중해야 합니다.

8 기능 설명

8.1 주요 기능

VS1003은 독점 디지털 신호 프로세서인 VS_DSP를 기반으로 합니다. 여기에는 MP3, WMA 및 WAV PCM + ADPCM 오디오 디코딩, MIDI 합성기에 필요한 모든 코드 및 데이터 메모리가 직렬 인터페이스, multi-rate 스테레오 오디오 DAC 및 아날로그 출력 증폭기 및 필터와 함께 포함되어 있습니다. 또한 ADPCM 오디오 인코딩은 마이크 증폭기와 A/D 변환기를 사용하여 지원됩니다. UART는 디버깅 목적으로 제공됩니다.

8.2 지원되는 오디오 코덱

컨벤션	
마크 설명	
+	형식이 지원됩니다
-	형식이 존재하지만 지원되지 않음
	형식이 존재하지 않습니다.

8.2.1 지원되는 MP3(MPEG 레이어 III) 형식

MPEG 1.0 1 :

샘플레이트/Hz	전송률 / kbit/s															
	32	40	48	56	64	80	96	112	128	160	192	224	256	320		
48000 +	+	+	+	+	+	+++				+++			+	++		
44100 +	+++			+	+	+	+	+	+	+++			+	++		
32000 +	+++			+	+	+	+	+	+	+++			+	++		

MPEG 2.0 1 :

샘플레이트/Hz	비트레이트/kbit/s															
	8	16	24	32	40	48	56	64	80	96	112	128	144	160		
24000 +	+++				+	+	+	+	+	+++			+	++		
22050 +	+++				+	+	+	+	+	+++			+	++		
16000 +	+++				+	+	+	+	+	+++			+	++		

MPEG 2.5 12 :

샘플레이트/Hz	비트레이트/kbit/s															
	8	16	24	32	40	48	56	64	80	96	112	128	144	160		
12000 + 11025	+++				+	+	+	+	+	+++			+	++		
+	+++				+	+	+	+	+	+++			+	++		
8000 +	+++				+	+	+	+	+	+++			+	++		

1 또한 모든 가변 비트 전송률(VBR) 형식이 지원됩니다.

2 MPEG 2.5는 표준 형식이 아니기 때문에 비호환성이 발생할 수 있습니다.

8.2.2 지원되는 WMA 형식

Windows Media 오디오 코덱 버전 2, 7, 8 및 9가 지원됩니다. 모든 WMA 프로필(L1, L2 및 L3)이 지원됩니다. 이전에는 스트림이 클래스 1: 2a, 2b 및 3으로 분리되었습니다. WMA 9 Professional 및 WMA 9 Lossless는 지원되지 않습니다. 디코더는 Microsoft의 적합성 테스트 프로그램을 통과했습니다.

WMA 4.0/4.1:

샘플레이트	전송률 / kbit/s																	
	/헤르츠	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000 + ++																		
11025			+	+														
16000				+	+	+	+	+										
22050							+	+++										
32000									+++				+	++				
44100											+		+++			+++		
48000																	++	

WMA 7:

샘플레이트	전송률 / kbit/s																	
	/헤르츠	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000 +		+	+			+												
11025			+	+														
16000				+	+	+	+	+										
22050						+	+++											
32000								+			+	+	+	+				
44100											+		+++			+++		+
48000																	++	

WMA 8:

샘플레이트	전송률 / kbit/s																	
	/헤르츠	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000 +		+	+			+												
11025			+	+														
16000					+++													
22050						+		+	+	+	+							
32000								+			+	+	+	+				
44100											+		+++			+++		+
48000																	++	+

WMA 9:

샘플레이트	전송률 / kbit/s																			
	/헤르츠	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192	256	320
8000 + ++		++				+														
11025			+	+																
16000				++																
22050						+++														
32000							+			+++										
44100								+		+			+++++							+
48000													+		+++					

이러한 예상 WMA 디코딩 프로필 외에도 가변 비트레이트 WMA 스트림을 포함하여 다른 모든 비트레이트 및 샘플레이트 조합이 지원됩니다. WMA는 MP3만큼 균등하게 비트스트림을 소비하지 않으므로 동일한 비트 전송률에서 깨끗한 재생을 위해서는 더 높은 피크 전송 기능이 필요합니다.

8.2.3 지원되는 RIFF WAV 형식

가장 일반적인 RIFF WAV 하위 형식이 지원됩니다.

형식 이름		지원되는 설명 16 및 8비트, 모든 샘플	
0x01	PCM	+	플 속도 < 48kHz
0x02 ADPCM	0x03	-	
	IEEE 플로트 0x06	-	
ALAW		-	
0x07 물라		-	
0x10	오키 ADPCM	-	
0x11	IT에는 ADPCM이 있습니다.	+	모든 샘플 속도 < 48kHz
0x15	디지스트D	-	
0x16	디지픽스	-	
0x30	돌비 AC2 0x31	-	
	GSM610	-	
0x3b 로크웰	ADPCM 0x3c 로크웰 디지털	-	
0x40		-	
	G721 ADPCM	-	
0x41	G728 CELP	-	
0x50 MPEG	0x55 0x64	-	
	MPEGLAYER3	+	지원되는 MP3 모드는 8.2.1장을 참조하십시오.
	G726 ADPCM	-	
0x65	G722 ADPCM	-	

8.2.4 지원되는 MLMI 형식

일반 MIDI 및 SP-MIDI 형식 0 파일이 재생됩니다. 형식 1 및 2 파일은 사용자가 형식 0으로 변환해야 합니다. 최대 동시 동시발음수는 40입니다. 실제 동시발음수는 내부 클럭 속도(사용자가 선택할 수 있음), 사용된 악기, 가능한 후처리 효과(예: 저음 및 고음 인핸서)에 따라 다릅니다. 폴리포니 제한 알고리즘은 SP-MIDI MIP 테이블(있는 경우)을 사용합니다.

36.86MHz(3.0x 입력 클록)는 16-26개의 동시 지속음을 달성합니다. 순간적인 음표의 양은 더 커질 수 있습니다. 36MHz는 전력 소비와 품질 간의 공정한 절충안이지만 더 높은 클록을 사용하여 다성음을 높일 수 있습니다.

VS1003b는 36개의 개별 계측기를 구현합니다. 각 멜로디, 효과 및 타악기는 이러한 악기 중 하나로 매핑됩니다.

VS1003b		
멜로디	효과	총격
비브라	리버스 심벌즈 베이스 드럼 기타 프렛 소음	
폰 피아노	스네어 호흡 폐쇄 hihat 해변 오픈 hihat	
오르	높은 톰 낮은 톰	
간 기		
타 왜곡 기타 베이	새 트윗 전화	
스 바이올린	헬리콥터 박	
	수 총성	크래쉬 심벌즈 2 라이
문자열		드 심벌즈 탬버린
트럼펫		
오른쪽		높은 콩가 낮은
플루		콩가
트 리드		마라카스
패드		열쇠
스틸드럼		

8.3 VS1003의 데이터 흐름

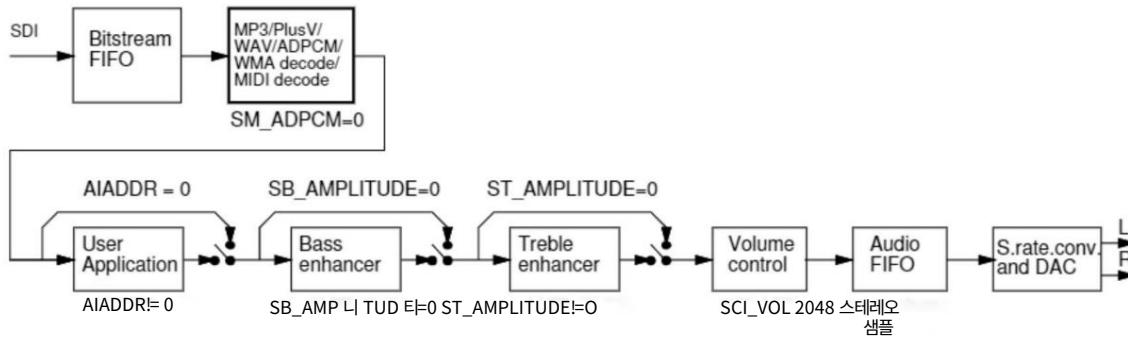


그림 16: VS1003의 데이터 흐름.

먼저 오디오 데이터에 따라 제공되는 ADPCM 인코딩 모드가 설정되지 않은 경우 MP3, WMA, PCM WAV, IMA ADPCM WAV 또는 MIDI 데이터가 SDI 버스에서 수신되어 디코딩됩니다.

디코딩 후, SCI_AIADDR이 0이 아니면, 해당 레지스터가 가리키는 주소에서 응용 프로그램 코드가 실행된다. 자세한 내용은 VS10XX용 애플리케이션 노트를 참조하십시오.

그런 다음 데이터는 SCI_BASS 레지스터에 따라 Bass 및 Treble Enhancer로 전송될 수 있습니다.

그런 다음 신호는 오디오 FIFO에 데이터를 복사하는 블루 헤더 장치로 공급됩니다.

오디오 FIFO는 오디오 인터럽트(10.13.1장)에서 읽고 샘플 속도 변환기와 DAC에 공급되는 데이터를 보유합니다. 오디오 FIFO의 크기는 2048 스테레오(2x16비트) 샘플 또는 8KiB입니다.

샘플 속도 변환기는 모든 다른 샘플 속도를 X7ALI/2 또는 사용 가능한 최고 샘플 속도의 128배로 변환합니다. 따라서 복잡한 PLL 기반 클록킹 방식이 필요하지 않으며 하나의 고정 입력 클록 주파수로 거의 무제한의 샘플 속도 정확도를 허용합니다. 12.288MHz 클록을 사용하는 DA 컨버터는 128 x 48kHz, 즉 6.144MHz에서 작동하고 스테레오 동위상 아날로그 신호를 생성합니다. 오버샘플링된 출력은 온칩 아날로그 필터에 의해 저역 통과 필터링됩니다. 이 신호는 이어폰 증폭기로 전달됩니다.

8.4 직렬 데이터 인터페이스(SDI)

직렬 데이터 인터페이스는 압축된 MP3 또는 WMA 데이터, WAV PCM 및 ADPCM 데이터 및 MIDI 데이터를 전송하기 위한 것입니다.

디코더의 입력이 유효하지 않거나 충분히 빠르게 수신되지 않으면 아날로그 출력이 자동으로 음소거됩니다.

또한 9장에서 설명한 것처럼 SDI를 통해 여러 다른 테스트를 활성화할 수 있습니다.

8.5 직렬 제어 인터페이스(SCI)

직렬 제어 인터페이스는 SPI 버스 사양과 호환됩니다. 데이터 전송은 항상 16비트입니다. VS1003은 인터페이스의 레지스터 쓰기 및 읽기로 제어됩니다.

제어 인터페이스의 주요 컨트롤은 다음과 같습니다.

- 작동 모드, 시계 및 내장 효과 제어
- 상태 정보 및 헤더 데이터에 대한 액세스
- 인코딩된 디지털 데이터에 대한 액세스
- 사용자 프로그램 업로드

8.6 SCI 레지스터

SCI 레지스터, 접두사 SCI						
등록 유형	재설정	0x0 rw	0x800	시간 1	약어[비트]	설명
0x1 rw	0x3C3	0 0 0 0 0 0 0		70 CLKI 1 모드		모드 제어
				40 CLKI 상태		VS1003 상태
0x2	rw			2100 클키 베이스		내장 베이스/트레블 인핸서
0x3	로우			11000 XTALI 5 클록 F		클록 주파수 + 승수
0x4	rw			40 CLKI DECODE TIME 디코딩 시간(초)		
0x5	rw			3200 CLKI 아우디타		기타 오디오 데이터 RAM
0x6	로우	0x7		80 CLKI 램		쓰기/읽기
	rw			80 CLKI WRAMADDR RAM의 기본 주소		쓰기/읽기
0x8	--	0		-HDAT0		스트림 헤더 데이터 0
0x9	--	0		-HDAT1		스트림 헤더 데이터 1
OxA	rw	0		3200 CLKI 2 AIADDR		신청서 시작 주소
OxB	rw	0		2100 CLKI 볼륨		볼륨 조절
OxC	rw	0		50 CLKI 2 AICTRL0		애플리케이션 제어 레지스터 0
황소자리	rw	0		50 CLKI 2 AICTRL1		애플리케이션 제어 레지스터 1
OxE	rw	0 0		50 CLKI 2 AICTRL2		애플리케이션 제어 레지스터 2
OxF	rw			50 CLKI 2 AICTRL3		애플리케이션 제어 레지스터 3

1 이 레지스터에 쓴 후 DREQ가 낮게 유지되는 최악의 시간입니다. 사용자는 실행하는 데 100 클럭 주기 미만이 소요되는 레지스터 쓰기에 대해 DREQ 검사를 건너뛰도록 선택할 수 있습니다.

2 또한 사용자 애플리케이션 루틴에서 소요되는 주기를 계산해야 합니다.

3 펌웨어는 이 레지스터의 값을 즉시 0x38로 변경하고 100ms 이내에 0x30으로 변경합니다.

4 모드 레지스터 쓰기가 소프트웨어 재설정을 지정하는 경우 최악의 경우 시간은 16600 XTALI 사이클입니다.

⁵ 이 레지스터에 기록하면 잠시 동안 내부 클럭이 $1.0 \times XTALI$ 로 실행될 수 있습니다. 따라서 그렇지 않다. 이 레지스터 업데이트가 진행되는 동안 SCI 또는 SDI 비트를 보내는 것이 좋습니다.

SCI 쓰기가 완료될 때 DREQ가 낮으면 SCI 쓰기 처리 후에도 DREQ가 낮게 유지됩니다.

8.6.1 SCI_MODE(RW)

SCI_MODE는 VS1003의 작동을 제어하는 데 사용되며 기본값은 0x0800(SM_SDINew 설정)입니다.

비트 이름	기능	값 설명	일반	동위상	오디오
0 SM_DIFF	미분	0 1		왼쪽 채널 반전됨 오른쪽 잘못됨 재설정 없음 재설정	
1 SM_SETTOZERO 0으로 설정		0 1			
2 에스엠_리셋	소프트 리셋	0 1			
3 SM_OUTOFWAV WAV 디코딩에서 점프		0 1	아니요 예	전	
4 SM_P다운	전력 다운	0 1		원 켜기 전원 끄기가 허용되지	
5 SM_TESTS	SDI 테스트 허용	0 1		않음 허용됨	
6 에스엠_스트림	스트림 모드	0 1	아니요 그래		
7 SM_SETTOZERO2 0으로 설정		0 1	맞다 틀리다	상	
8 SM_DACT	DCLK 활성 에지	0 1	승하다 하락하다		
9 SM_SDIORD	SDI 비트 순서	0 1	MSb 먼저 마지막 MSb		
10 SM_SDISHARE	SPI 칩 선택 공유	0 1	아니요 예		
11 에스엠_시뉴	VS10xx 기본 SPI 모드	0 1	아니요 예		
12 SM_ADPCM	ADPCM 녹화 활성	0 1	아니요 예		
13 SM_ADPCM_HP ADPCM 하이 패스 필터 활성		0 1	아니요 예	마	
14 이름	ADPCM 녹화 선택기	0 1	이크 라인 인		

SM_DIFF가 설정되면 플레이어는 왼쪽 채널 출력을 반전시킵니다. 스테레오 입력의 경우 이것은 가상 서라운드를 생성하고 모노 입력의 경우 차동 왼쪽/오른쪽 신호를 생성합니다.

SM_RESET을 1로 설정하면 소프트웨어 재설정이 시작됩니다. 이 비트는 자동으로 지워집니다.

중간에 WAV, WMA 또는 MIDI 파일의 디코딩을 중지하려면 SM_OUTOFWAV를 설정하고 SM_OUTOFWAV가 지워질 때까지 DREQ를 존중하는 데이터를 보내십시오. SCI_HDAT1도 지워집니다.

WMA 및 MIDI의 경우 스트림 전송을 계속하고 WAV의 경우 0을 전송하는 것이 가장 안전합니다.

비트 SM_PDOWN은 VS1003을 소프트웨어 전원 차단 모드로 설정합니다. 소프트웨어 전원 차단은 XRESET 핀으로 활성화된 하드웨어 전원 차단만큼 전력 효율적이지 않습니다.

SM_TESTS가 설정되면 SDI 테스트가 허용됩니다. 요미 테스트에 대한 자세한 내용은 Chapter 9.8을 참조하십시오.

SM_STREAM은 VS1003의 스트림 모드를 활성화합니다. 이 모드에서 데이터는 가능한 한 균일한 간격으로 전송되어야 하며(512바이트 미만의 데이터 블록이 바람직함) VS1003은 재생 속도를 최대 5%까지 변경하여 입력 버퍼를 절반으로 유지하려고 모든 시도를 합니다.

최상의 음질을 위해서는 평균 속도 오차가 0.5% 이내, 비트레이트가 160kbit/s를 초과하지 않아야 하며 VBR을 사용하지 않아야 합니다. 자세한 내용은 VS10XX용 애플리케이션 노트를 참조하십시오.
이 모드는 WMA 파일에서 작동하지 않습니다.

SM_DACT는 SDI용 데이터 클럭의 활성 에지를 정의합니다. '0'이면 상승 에지에서 데이터를 읽고 '1'이면 하강 에지에서 데이터를 읽습니다.

SM_SDIORD가 해제되면 SDI의 바이트가 먼저 기본 MSb로 전송됩니다. SM_SDIORD를 설정하면 사용자는 SDI의 비트 순서를 반대로 할 수 있습니다. 즉, 비트 0이 먼저 수신되고 비트 7이 마지막에 수신됩니다. 그러나 바이트는 여전히 기본 순서로 전송됩니다. 이 레지스터 비트는 SCI 버스에 영향을 미치지 않습니다.

SM_SDISHARE를 설정하면 SM_SDINEW도 설정된 경우 7.2장에서 설명한 것처럼 SCI와 SDI가 동일한 칩 선택을 공유합니다.

SM_SDINEW를 설정하면 장 7.2.1 및 7.4.2에 설명된 대로 VS1 Oxx 기본 직렬 모드가 활성화됩니다.
이 비트는 VS1003이 시작될 때 기본값으로 설정됩니다.

SM ADPCM과 SM RESET을 동시에 활성화하면 사용자는 IMA ADPCM 녹화 모드를 활성화합니다. 자세한 내용은 VS10XX용 애플리케이션 노트에서 확인할 수 있습니다.

SM_ADPCM_HP가 SM_ADPCM 및 SM_RESET과 동시에 설정되면 ADPCM 모드는 고역 통과 필터로 시작됩니다. 이는 배경 소음이 많을 때 말을 명료하게 하는 데 도움이 될 수 있습니다. ADPCM 엔코더 주파수 응답에 생성된 차이는 그림 17과 같습니다.

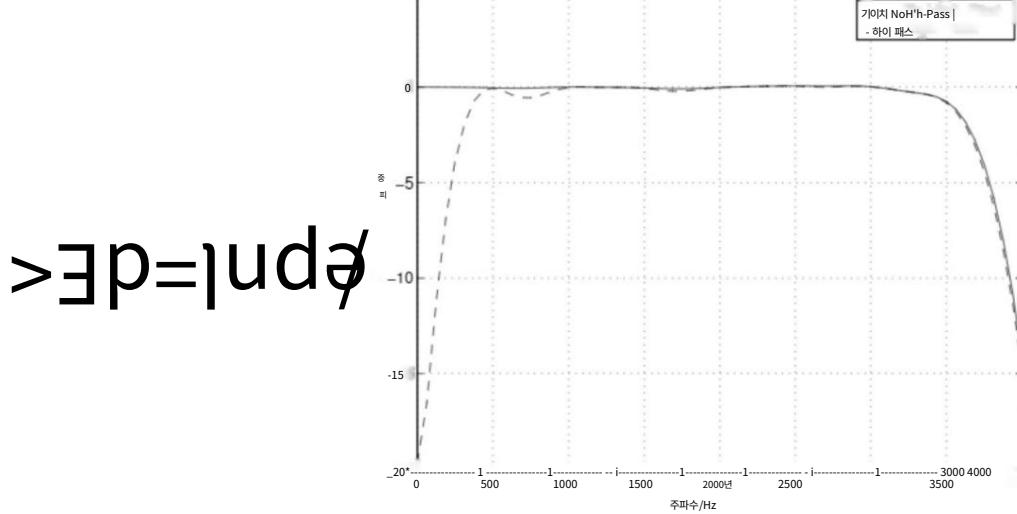


그림 17: 8kHz 샘플 속도의 ADPCM 주파수 응답.

SM_LINE_IN은 ADPCM 녹화를 위한 입력을 선택하는 데 사용됩니다. '0'이면 마이크 입력 핀 MICP 및 MICNOI 사용됩니다.
'1'이면 LINEINOI 사용됩니다.

8.6.2 SCI_STATUS(RW)

SCLSTATUS는 VS1003의 현재 상태에 대한 정보를 포함하고 사용자 종료를 허용합니다.

오디오 결함이 없는 칩.

이름	비트 설명
SS 버전	6:4 버전
SS_APDOWN2	3 아날로그 드라이버 전원 차단
SS_APDOWN1	2 아날로그 내부 전원 차단
SS_AVOL	1:0 아날로그 볼륨 컨트롤

SS_VER는 VS1001의 경우 0, VS1011의 경우 1, VS1002의 경우 2, VS1003의 경우 3입니다.

SS_APDOWN2는 아날로그 드라이버 전원 차단을 제어합니다. 일반적으로 이 비트는 시스템 펌웨어에 의해 제어됩니다. 그러나 사용자가 최소 전원 끄기 과도 상태로 VS1003의 전원을 끄려면 이 비트를 1로 설정한 다음 리셋을 활성화하기 전에 최소한 몇 밀리초 동안 기다리십시오.

SS_APDOWN1은 내부 아날로그 전원 차단을 제어합니다. 이 비트는 시스템 펌웨어에서만 사용됩니다.

SS_AVOL은 아날로그 볼륨 컨트롤입니다. 0 = -0dB, 1 = -6dB, 3 = -12dB. 이 레지스터는 시스템 펌웨어에 의해서만 자동으로 사용됩니다.

8.6.3 SCI_BASS(RW)

이름	비트	설명 ST
AMPLITUDE 15:12	1.5dB 단계의 고음 제어(-8..7, 0 = 꺼짐)	
ST_FREQLIMIT SB	11:8 1000Hz 단계의 하한 주파수(0..15)	
AMPLITUDE 7:4	1dB 단계로 저음 향상(0..15, 0 = 꺼짐)	
SB 주파수 제한	3:0 10Hz 단계의 하한 주파수(2..15)	

Bass Enhancer VSBE는 클리핑을 일으키지 않고 사용자 이어폰을 최대한 활용하는 강력한 베이스 부스팅 DSP 알고리즘입니다.

VSBE는 SB_AMPLITUDE가 0이 아닐 때 활성화됩니다. SB_AMPLITUDE는 사용자의 기본 설정으로 설정하고 SB_FREQLIMIT는 사용자의 오디오 시스템이 재생할 수 있는 최저 주파수의 약 1.5배로 설정해야 합니다. 예를 들어 SCI_BASS를 0x00f6으로 설정하면 60Hz 미만에서 15dB 향상됩니다.

참고: VSBE는 클리핑을 피하려고 하기 때문에 역동적인 음악 자료를 사용하거나 재생 볼륨이 최대로 설정되지 않은 경우 최상의 베이스 부스트를 제공합니다. 또한 베이스를 생성하지 않습니다. 소스 자료에는 처음부터 약간의 베이스가 있어야 합니다.

Treble Control VSTC는 ST_AMPLITUDE가 0이 아닌 경우 활성화됩니다. 예를 들어 SCI_BASS를 0x7a00으로 설정하면 10kHz 이상에서 10.5dB 고음이 향상됩니다.

Bass Enhancer는 44100Hz 샘플 속도에서 약 3.0 MIPS 및 Treble Control 1.2 MIPS를 사용합니다.
둘 다 동시에 켜질 수 있습니다.

8.6.4 SCI_CLOCKF(RW)

SCI_CLOCKF의 작동은 VS1001, VS1011, VS1002와 VS1003에서 다릅니다.

SCI_CLOCKF 비트		
이름	비트	설명
SC_MULT 15:13	클럭 승수	
SC_ADD 12:11	승수 추가 허용	
SC_FREQ 10:0	클럭 주파수	

SC_MULT는 내장 클럭 승수를 활성화합니다. 이것은 XTALI를 곱하여 더 높은 CLKI를 생성합니다.
값은 다음과 같습니다.

SC_MULT 마스크 CLKI	
0	0x0000 엑스탈리
1	0x2000 XTALI x 1.5 0x4000
2	XTALI x 2.0 0x6000 XTALI
3	x2.5 0x8000 XTALI x3.0
4	0xa000 XTALI x3.5
5	0xc000 XTALI x4.0 0xe000
6	XTALI x4.5
7	

SC_ADD는 WMA 스트림을 디코딩하는 데 mAE 주기가 일시적으로 필요한 경우 디코더 펌웨어가 SC_MULT에 지정된 승수에 추가할 수 있는 양을 알려줍니다. 값은 다음과 같습니다.

SC_ADD MASK 승수 추가	
0	0x0000 수정이 허용되지 않음
1	0x0800 0.5x
2	0x1000 1.0x
3	0x1800 1.5x

SC_FREQ는 입력 클럭 XTALI가 12.288MHz가 아닌 다른 값에서 실행 중인지 알려주는 데 사용됩니다.
XTALI는 4kHz 단계로 설정됩니다. 이 레지스터의 올바른 값을 계산하는 공식은 $\lceil \frac{f_{XTALI}}{4} \rceil$ (XTALI 단위는 Hz)입니다.

참고: 기본값 0은 X7ALI=12.288MHz를 의미하는 것으로 가정합니다.

참고: 최대 샘플링 속도가 $X1^j l < 12.288\text{MHz}$ 이기 때문입니다. — , XTALI인 경우 모든 샘플 속도를 사용할 수 없습니다.

참고: 자동 시계 변경은 WMA 파일을 디코딩할 때만 발생할 수 있습니다. 자동 시계 변경은 한 번에 0.5x씩 수행됩니다. 이로 인해 1.0x 클럭으로 떨어지지 않으며 WMA 파일 전체에서 동일한 SCI 및 SDI 클럭을 사용할 수 있습니다. 디코딩이 끝나면 기본 멀티플라이어가 복원되고 1.0x 클럭이 일시적으로 사용될 수 있습니다.

예: SCI_CLOCKF가 0x9BE8이면, SC_MULT = 4, SC_ADD = 3, SC_FREQ = 0x3E8 = 1000.

이는 $XTALI = 1000 \times 400() + 80000() = 12\text{MHz}$ 임을 의미합니다. 클록 승수는 $3.0 \times XTALI = 36\text{MHz}$ 로 설정되며 펌웨어가 자동으로 사용하도록 선택할 수 있는 최대 허용 승수는 $(3.0 + 1.5) \times XTALI = 54\text{MHz}$ 입니다.

8.6.5 SCI_DECODE_TIME(RW)

올바른 데이터를 디코딩할 때 현재 디코딩된 시간이 이 레지스터에 전체 초 단위로 표시됩니다.

사용자는 이 레지스터의 값을 변경할 수 있습니다. 이 경우 새 값을 두 번 써야 합니다.

SCI_DECODE_TIME은 모든 소프트웨어 재설정 시 및 WAV(PCM 또는 IMA AD PCM), WMA 또는 MIDI 디코딩이 시작되거나 종료될 때 재설정됩니다.

8.6.6 SCI_AUDATA(RW)

올바른 데이터를 디코딩할 때 현재 샘플 속도와 채널 수는 각각 SCI_AUDATA의 비트 15:1과 0에서 찾을 수 있습니다. 비트 15:1에는 샘플 속도를 2로 나눈 값이 포함되며 비트 0은 모노 데이터의 경우 0이고 스테레오의 경우 1입니다. SCI_AUDATA에 쓰면 샘플 속도가 직접 변경됩니다.

참고: 버그로 인해 홀수 샘플 속도는 VS1003b의 스테레오 비트 작동을 반전시킵니다.

예: 44100Hz 스테레오 데이터는 0xAC45(44101)로 읽습니다.

예: 11025Hz 모노 데이터는 0x2B10(11025)으로 읽습니다.

예: 11025Hz 스테레오 데이터는 0x2B11(11026)로 읽습니다.

예: 0xAC80을 쓰면 샘플 속도가 44160Hz로 설정되고 스테레오 모드는 변경되지 않습니다.

8.6.7 SCI_WRAM(RW)

SCI_WRAM은 애플리케이션 프로그램과 데이터를 명령 및 데이터 RAM에 업로드하는 데 사용됩니다.

시작 주소는 SCI_WRAM의 첫 쓰기/읽기 이전에 SCI_WRAMADDR에 쓰기로 초기화되어야 한다. 하나의 SCI_WRAM 쓰기/읽기로 16비트의 데이터를 전송할 수 있고 명령어의 길이가 32비트이므로 각 명령어에 대해 두 번의 연속적인 쓰기/읽기가 필요합니다. 바이트 순서는 빅 엔디안입니다(즉, 가장 중요한 단어가 먼저). 각 전체 단어 쓰기/읽기 후에 내부 포인터가 자동으로 증가합니다.

8.6.8 SCI_WRAMADDR(W)

SCI_WRAMADDR은 다음 SCI_WRAM 쓰기/읽기에 대한 프로그램 주소를 설정하는 데 사용됩니다.

X에는 주소 오프셋 0, Y에는 0x4000, 명령어 메모리에는 0x80000이 사용됩니다. 주변 레지스터에도 액세스할 수 있습니다.

SM_WRAMADDR 시작... 종료	목적지 주소 시작... 끝	비트/ 단어	설명
0x1800... 0x187F	0x1800... 0x187F	16 X 데이터	RAM
0x5800... 0x587F	0x1800... 0x187F	16 Y 데이터	RAM 32 16
0x8030... 0x84FF	0x0030... 0x04FF		명령 RAM
0xC000...0xFFFF	0xC000...0xFFFF		I/O

X, Y 및 명령 메모리의 사용자 영역만 위에 나열되어 있습니다. 다른 영역에 액세스할 수 있지만 달리 지정하지 않는 한 쓰지 않아야 합니다.

8.6.9 SCI_HDAT0 및 SCI_HDAT1(R)

WAV 파일의 경우 SCI_HDAT0 및 SCI_HDAT1은 각각 0x7761 및 0x7665로 읽습니다.

WMA 파일의 경우 SCI_HDAT1에는 0x574D가 포함되고 SCI_HDAT0에는 초당 바이트로 측정된 데이터 속도가 포함됩니다. 파일의 비트 전송률을 얻으려면 SCI_HDAT0 값에 8을 곱하십시오.

MIDI 파일의 경우 SCI_HDAT1은 0x4D54를 포함하고 SCI_HDAT0은 다음 표에 따른 값을 포함합니다:

HDAT 아 15:8] HDAT 아 7: 0] Value Explanation polyphony reserved			
0			현재 폴리포니
1..255			

MP3 파일의 경우 SCI_HDAT[0... 1]에는 다음 내용이 있습니다:

조금	기능	값 설명
HDAT1[15:5] 싱크워드		2047 스트림 유효
HDAT1[4:3]	ID	3 ISO 11172-3 MPG 1.0 ISO 2 13818-3 MPG 2.0(1/2 속도) 1 MPG 2.5(1/4 속도) 0 MPG 2.5(1/4율) 3 2
HDAT1[2:1]	총	
		II
		III
		0 예약됨 1
HDAT1 [0]	보호 비트	NoCRC
		0 CRC 보호
HDAT0[15:12] 비트레이트		ISO 11172-3 예약
HDAT0[11:10] 샘플 속도	3	됨 32/16/8kHz
	2	48/24/12kHz
	1 0	44/22/11kHz
HDAT0[9]	패드 비트	추가 슬롯 일반 프레
		1 0 임이 정의되지 않음
HDAT0[8] 전용 비트		
HDAT 아 7:6】 mode	3	단핵종
	2	듀얼 채널 조인트
	1	스테레오 스테레오
	0	
HDAT 아 5:4】 extension		ISO 11172-3 저작
HDAT0[3] 저작권	1	권이 있는 무료 원
	0	본
HDAT0[2]	원래의	1
		0 복사
HDAT 아 1:0] emphasis	3	CCITT J.17 예
	2	약됨
		50/15마이크로초
	1 0	없음

읽을 때 SCI_HDAT0 및 SCI_HDAT1에는 현재 디코딩 중인 MP3 스트림에서 추출한 헤더 정보가 포함됩니다. 재설정 후 두 레지스터가 모두 삭제되어 아직 데이터가 발견되지 않았음을 나타냅니다.

SCI_HDAT0의 Hsample rate 5 * 필드는 다음 표에 따라 해석됩니다:

"샘플 속도" 3	ID=3/Hz	ID=2/Hz ID=0,1 / Hz
	-	-
2	32000	16000
	48000	24000
10	44100	22050
		11025

HDATO의 "비트 전송률" 필드는 다음 표에 따라 읽습니다.

"비트 전송률"	ID=3/kbit/s	아이디=0,1,2/kbit/s
15	금지된 320	금지된 160
14		
13	256	144
12	224	128
11	192	112
10	160	96
9	128	80
8	112	64
7	96	56
6	80	48
5	64	40
4	56	32
3	48	24
2	40	16
1	32	8
0	-	-

8.6.10 SCI_AIADDR(RW)

SCI_AIADDR은 SCI_WRAMADDR 및 SCI_WRAM 레지스터로 이전에 작성된 응용 프로그램 코드의 시작 주소를 나타냅니다. 애플리케이션 코드를 사용하지 않는 경우 이 레지스터를 초기화하지 않거나 0으로 초기화해야 합니다. 자세한 내용은 VS10XX용 애플리케이션 노트를 참조하십시오.

8.6.11 SCI_VOL(RW)

SCI_VOL은 플레이어 하드웨어의 볼륨 컨트롤입니다. 각 채널에 대해 0..254 범위의 값을 정의하여 최대 볼륨 레벨(0.5dB 단계)에서 감쇠를 설정할 수 있습니다.

그런 다음 왼쪽 채널 값에 256을 곱하고 값을 더합니다. 따라서 최대 볼륨은 0이고 전체 무음은 OxFFE입니다.

예: 왼쪽 채널의 볼륨이 -2.0dB이고 오른쪽 채널이 -3.5dB인 경우: $(4 * 256) + 7 = 0x407$. 시작 볼륨은 전체 볼륨으로 설정되어 있습니다. 소프트웨어를 재설정해도 볼륨 설정은 재설정되지 않습니다.

참고: SCI_VOL을 OxFFFF로 설정하면 아날로그 전원 차단 모드가 활성화됩니다.

8.6.12 SCI_AICTRL[x] (RW)

SCI_AICTRL[x] 레지스터($x=[0 .. 3]$)는 니저의 응용 프로그램에 접근하는데 사용될 수 있다.

9 작동

9.1 클럭킹

VS1003은 단일 공칭 12.288MHz 기본 주파수 마스터 클록에서 작동합니다. 이 클록은 외부 회로(핀 X1Ani에 연결됨) 또는 내부 클록 크리스탈 인터페이스(핀 X7ALI 및 X7ALO)에 의해 생성될 수 있습니다.

9.2 하드웨어 리셋

XRESET 신호가 로우로 구동되면 VS1003이 재설정되고 모든 제어 레지스터와 내부 상태가 초기 값으로 설정됩니다. XRESETsignal은 모든 외부 클록과 비동기식입니다. 리셋 모드는 VS1003의 디지털 및 아날로그 부분이 모두 최소 전력 소비 단계에 있고 클럭이 중지되는 풀 파워 다운 모드의 두 배입니다. 또한 X7ALO는 접지되어 있습니다.

하드웨어 재설정 후(또는 전원을 켜 때) DREQ는 최소 16600 클럭 주기 동안 다운 상태를 유지합니다. 이는 VS1003이 12.288MHz에서 실행되는 경우 약 1.35ms 지연을 의미합니다. 그런 다음 사용자는 디코딩을 시작하기 전에 SCI_MODE, SCI_BASS, SCI_CLOCKF 및 SCI_VOL과 같은 기본 소프트웨어 레지스터를 설정해야 합니다. 자세한 내용은 섹션 8.6을 참조하십시오.

내부 클럭에 PLL을 곱할 수 있습니다. SCI_CLOCKF 레지스터를 통해 지원되는 승수는 1.() x .. .4.5x 입력 클럭이다. Internal Clock Multiplier의 재설정 값은 1.0x입니다. 일반적인 값이 필요한 경우 재설정 후 내부 클럭 승수를 3.0x로 설정해야 합니다. DREQ가 상승할 때까지 기다렸다가 SCI_CLOCKF(레지스터 3)에 값 0x9800을 기록한다. 자세한 내용은 섹션 8.6.4를 참조하십시오.

9.3 소프트웨어 재설정

경우에 따라 디코더 소프트웨어를 재설정해야 합니다. 이것은 SCI_MODE 레지스터의 비트 2를 활성화함으로써 이루어진다(8.6.1장). 그런 다음 최소 2 //초 동안 기다린 다음 DREQ를 확인합니다. DREQ는 최소 16600 클록 사이클 동안 다운 상태를 유지합니다. 이는 VS1003이 12.288MHz에서 실행되는 경우 대략 1.35ms 지연을 의미합니다. DREQ가 실행되면 평소와 같이 계속 재생할 수 있습니다.

VS1003이 낮은 비트 전송률 데이터 스트림의 끝을 자르지 않도록 하고 소프트웨어 재설정을 수행하려는 경우 파일 이후 및 재설정 전에 SDI 버스에 2048개의 0(DREQ 준수)을 공급하는 것이 좋습니다.. SCI_HDAT1 플링을 사용할 수도 있지만 이것은 MIDI 파일에 특히 중요합니다.

중간에 WAV, WMA 또는 MIDI 파일 재생을 중단하려면 모드 레지스터에 SM_OUTOFWAV를 설정하고 소프트웨어 재설정을 계속하기 전에 SCI_HDAT1이 자워질 때까지(2초 타임아웃 포함) 기다리십시오. MP3는 스트림 형식이기 때문에 현재 SM OUTOFWAV를 구현하지 않으므로 시간 초과 요구 사항이 있습니다.

9.4 ADPCM 기록

이 장에서는 IMA ADPCM 형식으로 RIFF/WAV 파일을 만드는 방법을 설명합니다. 이것은 널리 지원되는 ADPCM 형식이며 많은 PC 오디오 재생 프로그램에서 재생할 수 있습니다. IMA ADPCM 녹음은 선형 16비트 오디오에 비해 대략 4:1의 압축률을 제공합니다. 이를 통해 32.44kbit/s에서 8kHz 오디오를 녹음할 수 있습니다.

9.4.1 ADPCM 모드 활성화

IMA ADPCM 기록 모드는 SCI_MODE에서 SM_RESET 및 SM_ADPCM 비트를 설정하여 활성화됩니다. 선택적으로 SM_ADPCM_HP를 동시에 설정하여 8kHz 샘플 속도에 대해 고역 통과 필터를 활성화할 수 있습니다. SM_LINE_IN이 설정되면 마이크 대신 라인 입력이 사용됩니다.

ADPCM 기록을 활성화하기 전에 사용자는 SCI_AICTRL0에 클럭 분배기 값을 쓰고 SCI_AICTRL1에 게인해야 합니다.

SM_ADPCM_HP 사용의 차이점은 그림 17(30페이지)에 나와 있습니다. 일반적으로 SM_ADPCM_HP를 사용하지 않으면 오디오가 더 풍부해지고 원본에 더 가까워집니다. 그러나 고역 통과 필터를 활성화하면 말을 더 잘 이해할 수 있습니다. 8kHz 샘플 속도에서만 필터를 사용하십시오.

ADPCM 기록을 활성화하기 전에 사용자는 SCI_AICTRL0에 클럭 분배기 값을 써야 합니다.

샘플링 주파수는 다음 공식으로 계산됩니다: $f = \frac{f_c}{2^{d+1}}$. 여기서 내부 클럭(CLK)이고 d 는 SCI_AICTRL0의 분배기 값입니다. d 의 가장 낮은 유효한 값은 4이다. SCI_AICTRL0의 0을 담고 있다면, 기본 분할자 값 12가 사용된다.

예: $F_c =$

$$2.0 \times 12.288\text{MHz}, d = 12. \text{ 이제 } f = \frac{2.0}{2^{12+1}} = 8000 \text{ Hz.}$$

$$= 2.5 \times 14.745\text{MHz}, d = 18. \text{ 이제 } f = \frac{2.5}{2^{18+1}} = 25000 \text{ Hz.}$$

$$= 2.5 \times 13\text{MHz}, d = 16. \text{ 이제 } f = 25000 \text{ Hz.}$$

또한 ADPCM 모드를 활성화하기 전에 사용자는 SCI_AICTRL1 레지스터에 선형 기록 이득 제어를 설정해야 합니다. 1024는 디지털 게인과 같고 1,512는 디지털 게인 0.5와 같습니다. 사용자가 자동 이득 제어(AGC)를 사용하려면 SCI_AICTRL1을 0으로 설정해야 합니다. 일반적인 음성 응용 프로그램은 일반적으로 AGC를 사용하는 것이 좋습니다. 녹음에서 비교적 균일한 음성 크기를 관리하기 때문입니다.

VS1033C 이후 SCI_AICTRL2는 최대 AGC 이득을 제어합니다. SCI_AICTRL2가 0이면, 최대 이득은 65535(64x), 즉 전체 범위가 사용된다. 이것은 이전 작업과 호환됩니다.

9.4.2 IMA ADPCM 데이터 읽기

IMA ADPCM 기록이 활성화된 후 SCI_HDAT0 및 SCI_HDAT1 레지스터에 새로운 기능이 있습니다.

IMA ADPCM 샘플 버퍼는 1024개의 16비트 워드입니다. 버퍼의 채우기 상태는 SCI_HDAT1에서 읽을 수 있습니다. SCI_HDAT1이 0보다 크면 SCI_HDAT0에서 최대한 많은 16비트 단어를 읽을 수 있습니다. 데이터를 충분히 빨리 읽지 않으면 버퍼가 오버플로되어 빈 상태로 돌아갑니다.

주: SCI_HDAT1 > 8960이면, 샘플을 읽기 전에 버퍼가 오버플로되고 지워질 때까지 기다리는 것이 더 나을 수 있습니다. 그렇게 하면 버퍼 앤디언을 피할 수 있습니다.

각 IMA ADPCM 블록은 128워드, 즉 256바이트입니다. 데이터 읽기를 중단하고 나중에 계속하려면 128단어 경계에서 중지하십시오. 이렇게 하면 전체 블록을 건너뛰고 인코딩된 스트림이 유효한 상태로 유지됩니다.

9.4.3 RIFF 헤더 추가

"IMA ADPCM 파일을 RIFF/WAV 파일로 만들려면 실제 데이터 앞에 헤더를 추가해야 합니다. 이 형식에서 2바이트 및 4바이트 값은 리틀 엔디안(가장 낮은 바이트 먼저)입니다.

파일 오프셋 필드 이름	크기 바이트	설명
영역	4 청크 ID	4 "리프"
4 청크 크기	4 F0 F1 F2 F3	파일 크기 - 8
8 형식	4 "웨이브"	
12화	4 "fmt" 4	
16 SubChunk1Size	0x14 0x0 0x0 0x0 20 2 0x11 0x0	
20 오디오 형식		IMA ADPCM의 경우 0x11
22 NumOfChannels	2 0x1 0x0	8kHz용 모노 사운드 0x1f40for8kHz
24 샘플레이트	4 R0 R1 R2 R3	운드 0xf400
28바이트 속도	4 B0 B1 B2 B3	0xfd7
32 블록정렬	2 0x0 0x1 2	0x100
샘플당 34비트	0x4 0x0 2 0x2	4비트 ADPCM 2
36바이트추가 데이터	0x0 2 0xf9 0x1	
38 엑스트라데이터	4 "사실" 4 0x4	블록당 샘플(505)
40 SubChunk2ID	0x0 0x0 0x0	
44화		4
48 NumOfSamples	4 S0 S1 S2 S3 4 "데이터"	
52 SubChunk3ID		
56화	4 D0 D1 D2 D3	데이터 크기(파일 크기-60)
60 블록	256	첫 번째 ADPCM 블록
316	...	추가 ADPCM 데이터 블록

n개의 오디오 블록이 있는 경우 표의 값은 다음과 같습니다.

$$F = nx 256 + 52$$

$$R = F_s \text{ (계산 방법은 9.4.1장 참조)}$$

F_s

$$505 S = nx 505. D = nx 256$$

기록할 양을 미리 알고 있는 경우 실제 데이터 전에 전체 헤더를 채울 수 있습니다. 단, 얼마만큼 녹음할지 모른다면 녹음이 끝난 후 헤더 크기 데이터 F, S, D를 채워야 한다.

ADPCM 블록의 128워드(256바이트)는 다음과 같이 SCI_HDAT0에서 읽고 파일에 쓴다. SCI_HDAT0의 상위 8비트는 파일의 첫 번째 바이트로 쓰여져야 하고 하위 8비트는 다음에 쓰여져야 한다. 이는 일부 16비트 마이크로컨트롤러의 기본 작동과 상반되며 이를 올바르게 수행하려면 각별한 주의가 필요할 수 있습니다.

올바른 방법으로 파일을 작성했는지 확인하는 방법은 각 256바이트 블록의 바이트 2와 3(첫 번째 바이트는 바이트 0으로 계산)을 확인하는 것입니다. 바이트 3은 항상 0이어야 합니다.

9.4.4 ADPCM 데이터 재생

IMA ADPCM 녹화를 재생하려면 9.4.3장에 설명된 대로 헤더가 있는 파일이 있어야 합니다. 이 경우 오디오 파일과 마찬가지로 SDI를 통해 ADPCM 파일을 제공하기만 하면 됩니다.

9.4.5 샘플 속도 고려 사항

IMA ADPCM 재생을 지원하는 VS1 Oxx 칩은 모든 샘플 속도로 ADPCM 파일을 재생할 수 있습니다. 그러나 일부 다른 프로그램에서는 IMA ADPCM 파일이 8000 또는 11025Hz와 같은 정확한 샘플 속도를 가질 것으로 예상할 수 있습니다. 또한 일부 프로그램이나 시스템은 8000Hz 미만의 샘플 속도를 지원하지 않습니다.

그러나 적절한 클럭이 없으면 정확한 8kHz 샘플 속도를 얻지 못할 수 있습니다. 12MHz 클록이 있는 경우 $2.0 \times 12\text{MHz}$ 및 $d = 12$ 로 얻을 수 있는 가장 가까운 샘플 속도는 $f = 7812.5\text{Hz}/z$ 입니다. 빈도 오차는 2.4%에 불과 하므로 동일한 파일을 PC에서도 재생하려면 헤더에 $f = 8(0)0//z$ 를 설정하는 것이 가장 좋을 수 있습니다. 이로 인해

샘플이 조금 더 빠르게 재생됩니다(59초에 1분이 재생됨).

그러나 절대적으로 필요한 경우가 아니면 여기에 설명된 방식으로 샘플 속도를 조정해서는 안 됩니다.

증가된 데이터 속도를 희생하면서 더 나은 품질을 원하는 경우 더 높은 샘플 속도(예: 16kHz)를 사용할 수 있습니다.

9.4.6 예제 코드

다음 코드는 VS1003b/VS1023에서 IMA ADPCM 인코딩을 초기화하고 데이터를 읽는 방법을 보여줍니다.

```
const unsigned char header[] =
{ 0x52, 0x49, 0x49, 0x46, 0x46, 0x1c, 0x10, 0x00, 0x00, 0x57, 0x41, 0x56,
  0x45, 0x66, 0x6d, 0x74, 0x20, /*|RIFF ...WAVEfmt 1*/ 0x14, 0x00, 0x00, 0x00, 0x11, 0x00, 0x01, 0x00,
  0x40, 0x1f, 0x00, 0x00, 0x75, 0x12, 0x00, 0x00, /*l..... . 큐...{E. ., 1*/ 0x00, 0x01, 0x04, 0x00, 0x02,
  0x00, 0xf9, 0x01, 0x66, 0x61, 0x63, 0x74, 0x04, 0x00, 0x00, 0x00, /*l..... .u. 사실.... 0x5c, 0x1f, 0x00, 0x00,
  0x64, 0x61, 0x74, 0x61, 0xe8, 0x0f, 0x00, 0x00
};

서명되지 않은 문자 db[512]; /* 디스크에 저장하기 위한 데이터 버퍼 */
```

제이*/

};

버전: 1.09, 2018-03-16

```

무효 RecordAdpcm1003(무효) {
    /* VS1003b/VS1033c ♦/
    u_int16 w = 0, idx = 0;

    * * *      /* 디스크의 여유 공간 확인 및 찾기 ♦/

    SetMp3Vol(0x1414);           /* 녹음 모니터 볼륨 ♦/
    WriteMp3SpiReg(SCI_BASS,
                    0); /* 저음/고음 비활성화 ♦/

    WriteMp3SpiReg(SCI_CL0CKF, 0x4430); /* 2.Ox 12.288MHz */ Wait(100); WriteMp3SpiReg(SCI_AICTRL0,
    12);                         /* 사업부 -> 12=8kHz 8=12kHz 6=16kHz */

    대기(100);
    WriteMp3SpiReg(SCI_AICTRL1, 0);      /* 자동 이득 ♦/
    대기(100);

    경우 (line_in) {
        WriteMp3SpiReg(SCI_MODE, 0x5804); /* 일반 SW 리셋 + 기타 비트 */
    } 또 다른 {
        WriteMp3SpiReg(SCI_MODE, 0x1804); /* 일반 SW 리셋 + 기타 비트 ♦/

    } for (idx=0; idx < sizeof(header); idx++) { /* 먼저 헤더 저장 */
        db[idx] = 헤더[idx];
    }
    /* 필요한 경우 비율 설정 ♦/
    /*db[24]=비율;*/
    /*db[25]=속도*8;*/

    /* 녹음 루프 */ while
    (recording_on) {
        하다{
            w = ReadMp3SpiReg(SCI_HDAT1); } 동안 (w < 256 ||
            w >= 896); /*◆ 512바이트를 사용할 수 있을 때까지 대기 ♦/

        동안 (idx < 512) {
            w = ReadMp3SpiReg(SCI_HDAT0);
            db [idx++] = w>>8;
            db[idx++] = 0xFF보다 w;
        }
        idx = 0;
        write_block(datasector++, db); /* 디스크에 출력 블록 쓰기 */
    }
    ...          /* WAV 헤더 정보 수정 ♦/
    ...          /* 그런 다음 FAT 정보 업데이트 ♦/
    ResetMP3();           /* 일반 재설정, 기본 설정 복원 */
    SetMp3Vol(볼륨);
}

```

9.5 SPI 부트

GPIO0 가 부팅 시 풀업 저항으로 1로 설정되면 VS1003은 외부 SPI 메모리에서 부팅을 시도합니다 .

SPI 부트는 다음 핀을 재정의합니다.

일반 모드 SPI 부팅 모드	
GPIO0	xCS
GPIO1	CLK
도대체	연기
GPIO2	된장

메모리는 16비트 주소(예: 최소 1KiB)가 있는 SPI 버스 직렬 EEPROM이어야 합니다.

VS1003에서 사용하는 직렬 속도는 공칭 12.288MHz 클록에서 245kHz입니다. 메모리의 처음 3바이트는 0x50, 0x26, 0x48이어야 합니다. 정확한 레코드 형식은 VS10XX용 애플리케이션 노트에 설명되어 있습니다.

9.6 재생/디코드

VS1003의 일반 동작 모드입니다. SDI 데이터가 디코딩됩니다. 디코딩된 샘플은 내부 DAC에 의해 아날로그 도메인으로 변환됩니다. 디코딩 가능한 데이터가 없으면 SCI_HDAT0 및 SCI_HDAT1이 0으로 설정되고 아날로그 출력이 음소거됩니다.

디코딩을 위한 입력이 없으면 VS1003은 유휴 모드(디코딩 시보다 전력 소모가 적음)로 들어가고 유효한 데이터에 대해 직렬 데이터 입력을 능동적으로 모니터링합니다.

중간에 소프트웨어를 재설정하지 않고도 모든 다른 형식을 연속해서 재생할 수 있습니다. 각 스트림 후에 최소 4개의 0을 보냅니다. 그러나 스트림 간에 소프트웨어 재설정을 사용하는 것은 깨진 파일을 방지하므로 여전히 좋은 생각일 수 있습니다. 이 경우 소프트웨어 리셋을 실행하기 전에 디코딩이 완료될 때까지(SCI_HDAT0 및 SCI_HDAT1이 0이 됨) 기다려야 합니다.

9.7 PCM 데이터 공급

VS1003은 WAV 파일 헤더를 전송하여 PCM 디코더로 사용할 수 있습니다. WAV 파일에서 전송된 길이가 0 또는 0xFFFFFFFF인 경우 VS1003은 무한정(또는 SM_OUTOFWAV가 설정될 때까지) PCM 모드를 유지합니다. 8비트 선형 및 16비트 선형 오디오는 모노 또는 스테레오에서 지원됩니다.

9.8 SDI 테스트

VS1003에는 사용자가 메모리 테스트, SCI 버스 테스트 및 여러 가지 사인파 테스트를 수행할 수 있는 몇 가지 테스트 모드가 있습니다.

모든 테스트는 비슷한 방식으로 시작됩니다. VS1003이 하드웨어 재설정되고 SM_TESTS가 설정된 다음 테스트 명령이 SDI 버스로 전송됩니다. 각 테스트는 4바이트 특수 명령 시퀀스와 4개의 0을 전송하여 시작됩니다. 순서는 아래에 설명되어 있습니다.

9.8.1 테스트

사인 테스트는 8바이트 시퀀스 0x53 OxEF 0x6E n 0 0 0 0으로 초기화됩니다. 여기서 n은 사용할 사인 테스트를 정의합니다. n은 다음과 같이 정의됩니다.

n 비트	
이름	비트 설명
Fs_idx	7:5 샘플링 속도 인덱스
에스	4:0 사인 스윕 속도

Fs_idx	FS
0	44100Hz
1	48000Hz
2	32000Hz
삼	22050Hz
4	24000Hz
5	16000Hz
6	11025Hz
7	12000Hz

출력할 사인의 주파수는 이제 $F = Fs \times$ 에서 계산할 수 있습니다.

$$\frac{S}{128}.$$

예: 사인 테스트는 값 126(0b01111110)으로 활성화됩니다. n을 구성 요소로 나누면 $Fs_idx = 06011 = 30$ 으로 $Fs = 22050 // 30$ 입니다. $S = 0b111110 = 30$, 따라서 최종 사인 주파수 $F = 220507 / 2 \times 30 = 33075Hz$.

$$\frac{30}{128}$$

사인 테스트를 종료하려면 시퀀스 0x45 0x78 0x69 0x74 0 0 0 0을 보냅니다.

참고: 사인 테스트 신호는 디지털 볼륨 컨트롤을 거치므로 채널을 개별적으로 테스트할 수 있습니다.

9.8.2 핀 테스트

핀 테스트는 8바이트 시퀀스 0x50 OxED 0x6E 0x54 0 0 0으로 활성화됩니다. 이 테스트는 칩 생산 테스트 전용입니다.

9.8.3 메모리 테스트

메모리 테스트 모드는 8바이트 시퀀스 0x4D OxEA 0x6D 0x54 0 0 0으로 초기화됩니다. 이 시퀀스 후에 500000 클럭 주기를 기다립니다. 결과는 SCI 레지스터 SCI_HDAT0에서 읽을 수 있으며 '1' 비트는 다음과 같이 해석됩니다.

비트 마스크 의미 15	
	0x8000 테스트 완료
14:7	미사용
6	0x0040 멕스 테스트 성공 0x0020 양호 I
5	RAM
4	0x0010 양호 Y RAM
삼	0x0008 좋은 X RAM
2	0x0004 좋은 I ROM
1	0x0002 좋은 Y ROM
0	0x0001 좋은 X ROM
	0x807f 모두 확인

메모리 테스트는 RAM 메모리의 현재 내용을 덮어씁니다.

9.8.4 SCI 테스트

Sci 테스트는 8바이트 시퀀스 0x53 0x70 OxEE n 0 0 0으로 초기화됩니다. 여기서 n은 테스트할 레지스터 번호입니다. 주어진 레지스터의 내용을 읽고 SCI_HDAT0에 복사합니다. 테스트할 레지스터가 HDAT0이면 결과는 SCI_HDAT 1에 복사된다.

예: n이 0이면 SCI 레지스터 0(SCI_MODE)의 내용이 SCI_HDAT0에 복사된다.

VS1003 레지스터 10개

10.1 이 장을 읽어야 하는 사람

사용자가 VS1003에 DSP 효과와 같은 자체 기능을 추가하고자 할 때 사용자 소프트웨어가 필요합니다.

그러나 대부분의 VS1003 사용자는 VLSI 솔루션 웹 사이트에서 소프트웨어 플러그인만 다운로드하는 사용자를 포함하여 자신의 코드 작성이나 이 장에 대해 걱정할 필요가 없습니다.

10.2 프로세서 코어

VS_DSP는 광범위한 다목적 프로세서 기능을 갖춘 16/32비트 DSP 프로세서 코어입니다. VLSI 솔루션의 무료 VSKIT 소프트웨어 패키지에는 VS_DSP 프로세서 코어용 어셈블리 언어 또는 확장 ANSI C 프로그램을 작성, 시뮬레이션 및 디버그하는 데 필요한 모든 도구와 문서가 포함되어 있습니다. VLSI 솔루션은 또한 완전한 디버그 기능을 위한 완전한 통합 개발 환경 VSIDE를 제공합니다.

10.3 VS1003 메모리 맵

VS1003*의 메모리 맵은 그림 18에 나와 있습니다.

10.4 SCI 레지스터

8.6장에 설명된 SCI 레지스터는 여기에서 0xC000..0xC00F 사이에서 찾을 수 있습니다. 이 레지스터에 대해 SCI_CHANGE라고 불리는 주소 0xC010에 하나가 있다.

SCI 레지스터, 접두사 SCI				
등록 유형	재설정 약어[비트]			설명
옥콜로	-	0	CHANGE[5:0]	마지막 SCI 액세스 주소.

SCI 변경 비트	
이름	비트 설명
공상 <u>과학</u> 쓰기	4 마지막 액세스가 쓰기 주기인 경우 1입니다.
SCI 채널 주소	마지막 액세스의 3:0 SPI 주소.

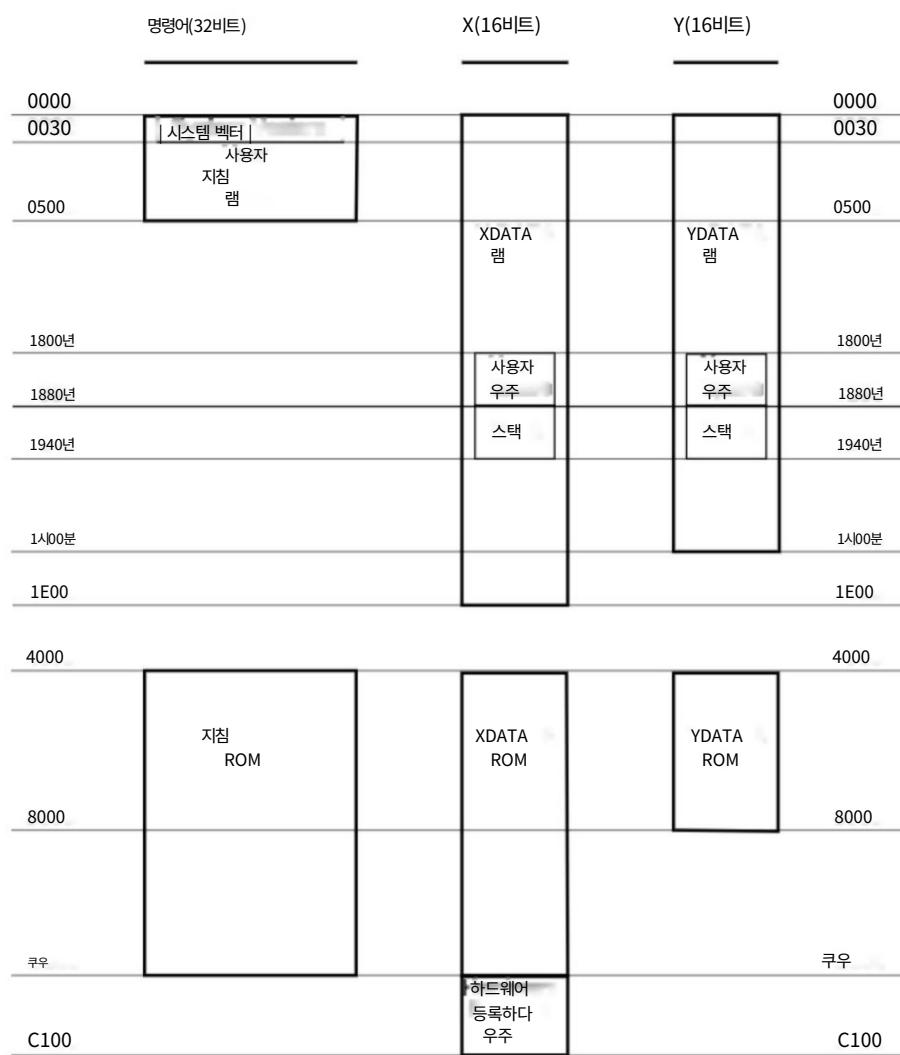


그림 18: 사용자의 메모리 맵.

10.5 직렬 데이터 레지스터

SDI 레지스터, 접두사 SER 설명 마지				
등록 유형 재설정 악어[비트]		막으로 수신된 2바		
0xC011	--	0	데이터	이트, 백엔디안.
0 의 0xC012			젠향[0]	DAMN 핀 컨트롤.

10.6 DAC 레지스터

DAC 레지스터, 접두사 DAC 설명 DAC			
등록 유형	재설정 약어[비트]		설명
0xC013 로우 0		FCTL	주파수 제어, 16 LSbs.
0xC014 로우 0 0xC015 로우 0		FCTLH	DAC 주파수 제어 4MSbs, PLL 제어.
		왼쪽	DAC 왼쪽 채널 PCM 값.
		오른쪽	DAC 우측 채널 PCM 값.

네 번째 클록 주기마다 내부 26비트 카운터가 $(\text{DAC_FCTLH} \& 15) \times 65536 + \text{DAC_FCTL}$ 에 의해 추가됩니다. 이 카운터가 오버플로될 때마다 `DAC_LEFT` 및 `DAC_RIGHT`의 값을 읽고 DAC 인터럽트를 생성합니다.

10.7 GPIO 레지스터

GPIO 레지스터, 접두사 GPIO		
등록 유형	재설정 약어[비트]	설명
0xC017 rw 0 0xC018 공	DDR[3:0]	방향.
장 0	IDATA[3: 0]	핀에서 값을 읽습니다.
0xC019 로우 0	ODATA[3: 0]	핀에 설정된 값을 설정합니다.

`GPIO_DIR`은 GPIO 핀의 방향을 설정하는 데 사용됩니다. 1은 출력을 의미합니다. `GPIO_ODATA`는 `GPIO_DIR` 비트가 입력으로 설정되어 있어도 해당 값을 기억합니다.

GPIO 레지스터는 인터럽트를 생성하지 않습니다.

VS1003에서 VSDSP 레지스터는 SCI_WRAMADDR 및 SCI_WRAM 레지스터를 통해 읽고 쓸 수 있습니다. 따라서 GPIO 핀을 매우 편리하게 사용할 수 있습니다.

10.8 인터럽트 레지스터

인터럽트 레지스터, 접두사 INT Reg Type		
Reset Abbrev[bits]		설명 ENABLE[7:0]
0xC01A 로우 0 0xC01B 0		인터럽트 활성화.
0xC01C 0 0xC01D 로우 0	글로벌 DIS[-]	인터럽트 카운터에 추가하기 위해 씁니다.
	글로벌 ENA[-]	interrupt 카운터에서 빼기 위해 씁니다.
	COUNIER[4: 이 인터럽트 카운터.	

INT_ENABLE은 인터럽트를 제어합니다. 제어 비트는 다음과 같습니다.

INT_ENABLE 비트	
이름	비트 설명 7 타이머 1 인터럽트
1시간1	트를 활성화합니다.
INT 1시간0	6 타이머 0 인터럽트를 활성화합니다.
INT 수신	5 UART RX 인터럽트를 활성화합니다.
텍사스에 있는 INT	4 UART TX 인터럽트를 활성화합니다.
INT와 MODU	3 AD 변조기 인터럽트를 활성화합니다.
SDI의 지능	2 데이터 인터럽트를 활성화합니다.
INT EN SCI	1 SCI 인터럽트를 활성화합니다.
DAC의 INT	0 DAC 인터럽트를 활성화합니다.

참고: INT_ENABLE을 변경하면 효과가 나타나기까지 최대 6클럭 주기가 걸릴 수 있습니다.

INT_GLOB_DIS에 값을 쓰면 인터럽트 카운터 INT_COUNTER에 1이 추가되고 모든 인터럽트가 효과적으로 비활성화됩니다. 이 레지스터에 쓰기가 적용되기 전에 최대 6클록 주기가 걸릴 수 있습니다.

INT_GLOB_ENA에 값을 쓰면 인터럽트 카운터에서 1을 뺍니다(INT_COUNTER가 이미 0이 아닌 경우). 인터럽트 카운터가 0이 되면 INT_ENABLE로 선택한 인터럽트가 복원됩니다. 인터럽트 루틴은 인터럽트가 자동으로 인터럽트 카운터에 1을 추가하기 때문에 항상 마지막 작업으로 이 레지스터에 기록해야 하지만 초기 값으로 다시 빼는 것은 사용자의 책임입니다. 이 레지스터를 쓰기 전에 최대 6클록 주기가 걸릴 수 있습니다.

INT_COUNTER를 읽어 사용자는 인터럽트 카운터가 올바른지 여부를 확인할 수 있습니다. 레지스터가 0이 아니면 인터럽트가 비활성화됩니다.

10.9 A/D 변조기 레지스터

인터럽트 레지스터, 접두사 AD				
등록 유형	재설정 약어[비트]			설명 0xC01E rw
0xC01F rw	0	DIV		A/D 변조기 분배기.
	0	데이터		A/D 변조기 데이터.

AD_DIV는 AD 변환기의 샘플링 주파수를 제어합니다. 하나의 샘플을 수집하기 위해 $128 \times n$ 클럭 주기가 사용됩니다 (n 은 AD_DIV의 값임). 사용 가능한 가장 낮은 값은 4이며 CLKI가 24.576MHz일 때 48kHz 샘플 속도를 제공합니다. AD_DIV가 0이면 A/D 변환기가 꺼집니다.

AD_DATA는 가장 최근에 디코딩된 A/D 값을 포함합니다.

10.10 위치독

위치독은 위치독 카운터와 일부 로직으로 구성됩니다. 재설정 후 위치독은 비활성화됩니다. 카운터 리로드 값은 WDOG_CONFIG에 기록하여 설정할 수 있습니다. 위치독은 WDOG_RESET을 등록하기 위해 0x4ea9를 작성하여 활성화됩니다. 이 작업을 수행할 때마다 위치독 카운터가 재설정됩니다. 65536번째 클록 주기마다 카운터가 1씩 감소합니다. 카운터가 언더플로되면 vsdsp의 내부 재설정 시퀀스가 활성화됩니다.

따라서 WDOG_RESET에 대한 첫 번째 0x4ea9 쓰기 후 동일한 값을 가진 동일한 레지스터에 대한 후속 쓰기는 매 65536x WDOG_CONFIG 클록 주기마다 이루어져야 합니다.

일단 시작되면 위치독을 끌 수 없습니다. 또한 WDOG_CONFIG에 대한 쓰기는 카운터 재로드 값을 변경하지 않습니다.

위치독이 활성화된 후 WDOG_CONFIG 또는 WDOG_DUMMY에 대한 모든 읽기/쓰기 작업은 WDOG_RESET에 대한 다음 쓰기 작업을 무효화합니다. 이렇게 하면 런어웨이 루프가 올바른 숫자를 쓰더라도 카운터를 재설정하는 것을 방지할 수 있습니다. WDOG_RESET에 잘못된 값을 쓰면 WDOG_RESET에 대한 다음 쓰기도 무효화됩니다.

위치독 레지스터에서 읽기는 정의되지 않은 값을 반환합니다.

10.10.1 레지스터

위치독, 접두사 WD0G			
등록	유형	재설정	설명
		0 CONFIG 구성	
0xC021	에서	0 리셋	시계 구성
0xC022	에서	0 DUMMY[-] 더미 레지스터	

10.11 UART(범용 비동기 수신기/송신기)

RS232 UART는 RS232 표준 8N1(8 데이터 비트, 패리티 없음, 1 정지 비트)을 사용하여 직렬 인터페이스를 구현합니다.



그림 19: RS232 직렬 인터페이스 프로토콜

라인이 유휴 상태이면 로직 하이 상태를 유지합니다. 바이트가 전송되면 전송은 시작 비트(논리 0)로 시작하여 데이터 비트(LSB 먼저)로 계속되고 종지 비트(로직 하이)로 끝납니다. 각 8비트 바이트 프레임에 대해 10비트가 전송됩니다.

10.11.1 레지스터

UART 레지스터, 접두사 UARTx		
등록 유형	재설정 약어	설명
0xC028	--	0 STATUS[3: 0] 상태
0xC029 r/w	0xC02A	0 DATA[7: 0] Data
r/w 0xC02B	r/w	0 DATAH[15:8] 데이터 높음
		0 DIV 분할기

10.11.2 상태 UARTx_STATUS

상태 레지스터의 읽기는 송신기 및 수신기 상태를 반환합니다.

UARTx 상태 비트	
이름	비트 설명
UART_ST_RXORUN	3 수신기 오버런
UART_ST_RXFULL	2 수신기 데이터 레지스터 가득 참
UART_ST_TXFULL	1 송신기 데이터 레지스터 가득 참
UART_ST_TXRUNNING	0 송신기 작동 중

UART_ST_RXORUN은 수신기 시프트 레지스터에서 데이터 레지스터로 전송될 때 수신된 바이트가 읽지 않은 데이터를 덮어쓰면 설정되고, 그렇지 않으면 지워집니다.

데이터 레지스터에 읽지 않은 데이터가 있으면 UART_ST_RXFULL이 설정됩니다.

UART_ST_TXFULL은 데이터 레지스터에 대한 쓰기가 허용되지 않는 경우(데이터 레지스터 가득 참) 설정됩니다.

송신기 시프트 레지스터가 작동 중인 경우 UART_ST_TXRUNNING이 설정됩니다.

10.11.3 데이터 UARTx_DATA

UARTx_DATA에서 읽기는 수신된 바이트를 비트 7:0으로 반환하고 비트 15:8은 '0'로 반환합니다. 더 이상 읽을 데이터가 없으면 수신기 데이터 레지스터 가득 참 표시기가 지워집니다.

수신기 시프트 레지스터에서 수신기 데이터 레지스터로 바이트가 이동할 때 수신 인터럽트가 생성됩니다.

UARTx_DATA에 쓰기는 전송을 위한 바이트를 설정합니다. 데이터는 비트 7:0에서 가져오고 기록된 값의 다른 비트는 무시됩니다. 송신기가 유휴 상태이면 바이트가 즉시 송신기 시프트 레지스터로 이동되고 전송 인터럽트 요청이 생성되고 전송이 시작됩니다. 송신기가 사용 중인 경우 UART_ST_TXFULL이 설정되고 이전 바이트가 전송되고 전송이 진행될 수 있을 때까지 바이트가 송신기 데이터 레지스터에 남아 있습니다.

10.11.4 데이터 높음 UARTx_DATAH

비트 15:8이 사용된다는 점을 제외하면 UARTx_DA7A와 동일합니다.

10.11.5 분배기 UARTx_DIV

UARTx DIV 비트	
이름	비트 설명 15:8 분할자
UART 사업부 D1	1(0..255)
UART 사업부 D2	7:0 분배기 2(6..255)

재설정 시 분배기는 0x0000으로 설정됩니다. ROM 부트 코드는 올바른 비트 속도를 얻기 위해 마스터 클록 주파수에 따라 올바르게 초기화해야 합니다. 두 번째 분할자 (Z2)는 6에서 255 사이여야 합니다.

통신 속도 $f = (nL \cdot f_{clk}) / 4$: 여기서 f 는 bps 단위의 TX/RX 속도입니다.

nL 마스터 클럭 주파수이고 f 는

26MHz 마스터 클록에서 일반적인 통신 속도에 대한 분배기 값:

UART 속도의 예, $f_{Tn} = 2^{16}$ Hz Comm. 속도		
[bps]	UART DIV D1	UART DIV D2
38400		63
	42	63
	42	42
	51	26
	42	21
	25	26
57600	1	226
115200	0	226

10.11.6 인터럽트와 연산

송신기는 다음과 같이 작동합니다. 8비트 워드가 전송 데이터 레지스터에 기록된 후 송신기가 이전 바이트를 전송하는 중이 아니면 즉시 전송됩니다. 전송이 시작되면 TXJNTR 인터럽트가 전송됩니다. 상태 비트[1]은 송신기 데이터 레지스터 비어 있음(또는 전체 상태)을 알리고 비트[0]은 송신기(시프트 레지스터) 비어 있음 상태를 알립니다. 비어 있지 않은 경우(비트 [1] = '0') 송신기 데이터 레지스터에 새 워드를 쓰지 않아야 합니다. 송신기 데이터 레지스터는 송신기로 이동되고 전송이 시작되는 즉시 비어 있게 됩니다. 전송 인터럽트가 생성될 때마다 송신기 데이터 레지스터에 새 단어를 쓰는 것이 안전합니다.

수신기는 다음과 같이 작동합니다. RX 신호 라인을 샘플링하고 높음에서 낮음으로의 전환을 감지하면 시작 비트를 찾습니다. 그런 다음 비트 시간 중간에 각 8비트를 샘플링하고(상수 타이머 사용) 수신기(시프트 레지스터) LSB를 먼저 채웁니다. 마지막으로 정지 비트(로직 하이)가 감지되면 수신기의 데이터가 수신 데이터 레지스터로 이동하고 RXJNTR 인터럽트가 전송되고 상태 비트[2](수신 데이터 레지스터 가득 참)가 설정되고 상태 비트[2] 이전 상태가 비트[3]에 복사됩니다(수신 데이터 오버런). 그 후 수신기는 유휴 상태로 돌아가서 새로운 시작 비트를 기다립니다. 상태 비트[2]는 수신기 데이터 레지스터를 읽을 때 0이 됩니다.

RS232 통신 속도는 2개의 클록 분배기를 사용하여 설정합니다. 기본 클럭은 프로세서 마스터 클럭입니다. 이 레지스터의 비트 15-8은 첫 번째 분할기용이고 비트 7-0은 두 번째 분할기용입니다. RX 샘플 주파수는 두 번째 분배기에 입력되는 클럭 주파수입니다.

10.12 타이머

서로 독립적으로 초기화 및 활성화할 수 있는 두 개의 32비트 타이머가 있습니다. 활성화된 경우 타이머는 프로세서가 기록한 시작 값으로 초기화되고 모든 클록 주기가 감소하기 시작합니다. 값이 0을 초과하면 인터럽트가 전송되고 타이머는 시작 값 레지스터의 값으로 초기화되고 계속해서 감소합니다. 타이머는 활성화되어 있는 한 해당 루프에 유지됩니다.

타이머에는 다운 카운팅을 위한 32비트 타이머 레지스터와 프로세서가 기록한 타이머 시작 값을 유지하기 위한 32비트 TIMER1_LH 레지스터가 있습니다. 타이머에는 2비트 TIMER_ENA 레지스터도 있습니다. 각 타이머는 활성화 레지스터의 해당 비트에 의해 활성화(1) 또는 비활성화(0)됩니다.

10.12.1 레지스터

타이머 레지스터, 접두사 TIMER		
등록 유형	재설정 약어	설명
0xC030 r/w	0xC031 r/w	CONFIGR: 이 Timer configuration
		0 ENABLE[1:0] 타이머 활성화
0xC034 r/w	0xC035	0 T0L TimerO 시작값 - LSB
r/w 0xC036 r/w		0시간 TimerO 시작값 - MSB
0xC037 r/w	0xC038	0 T0CNTL TimerO 카운터 - LSB
r/w 0xC039 r/w		0 T0CNTH TimerO 카운터 - MSB
0xC03A r/w	0xC03B	0 T1L 타이머 시작값 - LSB
r/w		0 T1H 타이머 시작값 - MSB
		0 T1CNTL 타이머 카운터 - LSB
		0 T1CNTH 타이머 카운터 - MSB

10.12.2 구성 타이머 구성

타이머 구성 비트	
이름	비트 설명
타이머 CF_CLKDIV	7:0 마스터 클록 분배기

TIMER_CF_CLKDIV는 모든 타이머 클록의 마스터 클록 분배기입니다. 생성된 내부 클록 주파수 $f_i = f / c$, 여기서 f 는 마스터 클록 주파수이고 c 는 TIMER_CF_CLKDIV입니다.

예: 12MHz 마스터 클록의 경우 TIMER_CF_DIV=3은 마스터 클록을 4로 나누므로 출력/샘플링 클록은 $f_i = 3MHz$ 가 됩니다.

10.12.3 구성 TIMER_ENABLE

타이머 활성화 비트	
이름	비트 설명
T1 시간	— — 1 타이머 1 활성화
T0 시간	— — 0 타이머 활성화 0

10.12.4 타이머 X 시작값 TIMER_Tx[L/H]

32비트 시작 값 TIMER_Tx[L/H]는 타이머가 재설정될 때 초기 카운터 값을 설정합니다. 타이머 인터럽트 주파수 $f =$ 여기서 f_i 는 클럭 분할기로 얻은 마스터 클럭입니다(10.12.2장 참조 및 c 는 TIMER_Tx[L/H]입니다).

예: 12MHz 마스터 클록과 TIMER_CF_CLKDIV=3, 마스터 클록 $f_i = 3MHz$. TIMER_TH=0, TIMER_TL=99이면 타이머 인터럽트 빈도 $f_t =$ 행 $\frac{f}{3} =$
 $30kHz$.

10.12.5 타이머 X 카운터 TIMER_TxCNT[L/H]

TIMER_TxCNT[L/H]는 현재 카운터 값을 포함합니다. 이 레지스터 쌍을 읽음으로써 사용자는 다음 타이머 인터럽트까지 걸리는 시간을 알 수 있습니다. 또한 이 레지스터에 기록함으로써 원샷 길이가 다른 타이머 인터럽트 지연을 실현할 수 있습니다.

10.12.6 인터럽트

각 타이머에는 자체 인터럽트가 있으며 타이머 카운터가 언더플로될 때 발생합니다.

10.13 시스템 벡터 태그

시스템 벡터 태그는 사용자가 여러 디코더 기능을 제어하기 위해 교체할 수 있는 태그입니다.

10.13.1 오디오, 0x20

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi DAC_INT_ADDRESS,(i6)+l
```

사용자는 마음대로 오디오 인터럽트에 대한 제어권을 얻기 위해 첫 번째 명령을 jmpi 명령으로 바꿀 수 있습니다.

10.13.2 스클립트, 0x21

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi SCI_INT_ADDRESS,(i6)+l
```

사용자는 마음대로 명령을 jmpi 명령으로 대체하여 SCI 인터럽트를 제어할 수 있습니다.

10.13.3 DataInt, 0x22

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi SDI_INT_ADDRESS,(i6)+l
```

사용자는 마음대로 명령을 jmpi 명령으로 대체하여 SDI 인터럽트를 제어할 수 있습니다.

10.13.4 모듈, 0x23

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi M0DU_INT_ADDRESS,(i6)+l
```

사용자는 원하는 대로 명령을 jmpi 명령으로 대체하여 AD 변조기 인터럽트를 제어할 수 있습니다.

10.13.5 TxInt, 0x24

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi EMPTY_INT_ADDRESS,(i6)+l
```

사용자는 UART TX 인터럽트에 대한 제어권을 얻기 위해 임의로 명령을 jmpi 명령으로 대체할 수 있습니다.

10.13.6 RxInt, 0x25

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi RX_INT_ADDRESS,(i6)+l
```

사용자는 마음대로 UART RX 인터럽트에 대한 제어권을 얻기 위해 첫 번째 명령을 jmpi 명령으로 바꿀 수 있습니다.

10.13.7 Timer0Int, 0x26

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi EMPTY.INT.ADDRESS,(i6)+l
```

사용자는 마음대로 타이머 0 인터럽트에 대한 제어권을 얻기 위해 첫 번째 명령을 jmpi 명령으로 바꿀 수 있습니다.

10.13.8 타이머 Int, 0x27

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

```
jmpi EMPTY_INT_ADDRESS,(i6)+l
```

사용자는 마음대로 타이머 1 인터럽트에 대한 제어권을 얻기 위해 첫 번째 명령을 jmpi 명령으로 바꿀 수 있습니다.

10.13.9 사용자 코덱, 0x0

일반적으로 다음 VS_DSP 어셈블리 코드를 포함합니다.

jrc
아니

사용자가 표준 디코더에서 제어권을 가져오려면 첫 번째 명령을 너저 자신의 코드에 적절한 j 명령으로 대체해야 합니다.

사용자가 MP3 또는 WMA 데이터를 동시에 공급하지 않는 한 시스템은 1ms 이내에 사용자 프로그램을 활성화합니다. 이후 사용자는 시스템에서 인터럽트 벡터를 훔쳐 사용자 프로그램을 삽입해야 합니다.

10.14 시스템 벡터 함수

시스템 벡터 함수는 사용자가 자신의 응용 프로그램을 구현하는 데 도움이 되도록 호출할 수 있는 일부 함수에 대한 포인터입니다.

10.14.1 WritelRam(), 0x2

VS_DSP C 프로토타입:

```
void WritelRam(register iO u_int16 *addr, register a1 u_int16 msW, register a0 u_int16 lsW);
```

이것은 사용자 명령 RAM에 쓰는 기본 방법입니다.

10.14.2 ReadlRamQ, 0x4

VS_DSP C 프로토타입:

```
u_int32 ReadlRam(레지스터 iO u_int16 *addr);
```

이것은 사용자 명령 RAM에서 읽기에 선호되는 방법입니다.

A1은 MSB를 포함하고 a0는 결과의 LSB를 포함합니다.

10.14.3 데이터바이트(), 0x6

VS_DSP C 프로토타입:

```
u_int16 DataBytes(void);
```

사용자가 자신의 코드를 가리키도록 사용자 코덱의 포인터를 전환하여 시스템의 정상적인 작동을 인계받은 경우 이 기능과 다음 두 가지 기능을 통해 데이터 인터페이스에서 데이터를 읽을 수 있습니다.

이 함수는 읽을 수 있는 데이터 바이트 수를 반환합니다.

10.14.4 GetDataByte(), 0x8

VS_DSP C 프로토타입:

```
u_int16 GetDataByte(보이드);
```

데이터 인터페이스에서 한 데이터 바이트를 읽고 반환합니다. 이 함수는 입력 버퍼에 충분한 데이터가 있을 때까지 기다립니다.

10.14.5 GetDataWords(), 육사

VS_DSP C 프로토타입:

```
무효 GetDataWords(등록 iO y u_int16 *d, 등록 aO u_int16 n);
```

n 데이터 바이트 쌍을 읽고 빅 엔디안 형식(MSB의 첫 번째 바이트)으로 d에 복사합니다. 이 함수는 입력 버퍼에 충분한 데이터가 있을 때까지 대기합니다.

10.14.6 재부팅(), 0xc

VS_DSP C 프로토타입:

```
무효 재부팅(무효);
```

소프트웨어를 재부팅합니다. 즉, I RAM 벡터를 다시 초기화하지 않고 표준 펌웨어로 점프합니다.

이것은 완전한 초기화를 유발하는 소프트웨어 재설정 기능과 동일하지 않습니다.



11 최신 문서 버전 변경 사항

이 장에서는 이 문서의 가장 중요한 최신 변경 사항에 대해 설명합니다.

버전 1.09, 2018-03-16

- 8N1 형식에 대한 언급이 10.11장, UART(Universal Asynchronous Re 수신기/송신기).
- 마지막 페이지에 칩 이미지를 추가했습니다.
- 모든 특허가 만료되었으므로 1장, 라이센스 및 기타 관련 위치에서 MP2 및 MP3 라이센스 설명을 제거 했습니다.
- 3장, 정의를 업데이트했습니다.
- 기타 사소한 변경.

버전 1.08, 2014-12-19

- 12장, 연락처 정보에 업데이트된 전화번호.

버전 1.07, 2014-03-11

- VS1003B-LK는 VS1003B-L 및 VS1003B-B와 동일한 보다 완화된 CVD 제한에 대해 인증되었습니다. 이로 인해 VS1003B-LK에 대 한 보다 엄격한 CVD 제한이 4.1장 절대 최대 정격 및 4.2장 권장 작동 조건에서 제거되었습니다.

버전 1.06, 2012-03-16

- 4장, 특성 및 사양에서 VS1003B-LK에 대한 CVD 제한이 추가되었습니다 .

버전 1.05, 2011-04-13

- SCI 테스트 설명이 수정되었습니다.
- CVD 제한이 변경되었습니다.

버전 1.04, 2009-02-03

- 섹션 4.7에 일반적인 특성이 추가되었으며 섹션 4.3에서 일부 값이 변경되었습니다.

12 연락처 정보

VLSI 솔루션 Oy
입구 G, 2층
헤르미안카투 8
FI-33720 탐페레
핀란드

URL: <http://www.vlsi.fi/>
전화: +358-50-462-3200 상업
이메일: sales@vlsi.fi

이 문서에 대한 기술 지원이나 제안은 <http://www.vsdsp-forum.com/>에 참여하십시오. 기밀 기술 논의는 support@vlsi.fi로 문의하십시오.

