 CONFIDENTIAL INFORMATION

Revision	ECO #	Description of Change	Date	Approval
00	836	Production Release	11/13/90	

WARNING:

THIS DOCUMENT CONTAINS 13 D SIZE SCHEMATICS THAT ARE VERY DETAILED. IT IS RECOMMENDED THAT YOU ONLY REVIEW UP TO PAGE 22 ON THE SCREEN AND PRINT PAGES 23 - 34. PRINTING WILL TAKE APPROX. 8 MINUTES.

Originator/Date	Title		Document #
Salvatore Arcuri 4/1/90	**Spec,Asic,Monitor**		**1162.00**
Department: Analog Engineering		Size: **A**	**Sheet 1 of 34**

1.0 Description

The monitor asic is a digital gate array to be used inside the monitor. In this document we shall refer to it as the asic. The purpose of the asic is to allow the computer to communicate with keyboard, mouse, sound out (DACs and attenuator) and soundin (Codec) devices. It is implemented in 1 micron triple layer metal CMOS technology and packaged in a 28 pin PLCC. It is manufactured by Motorola. A functional block diagram is given in figure1.

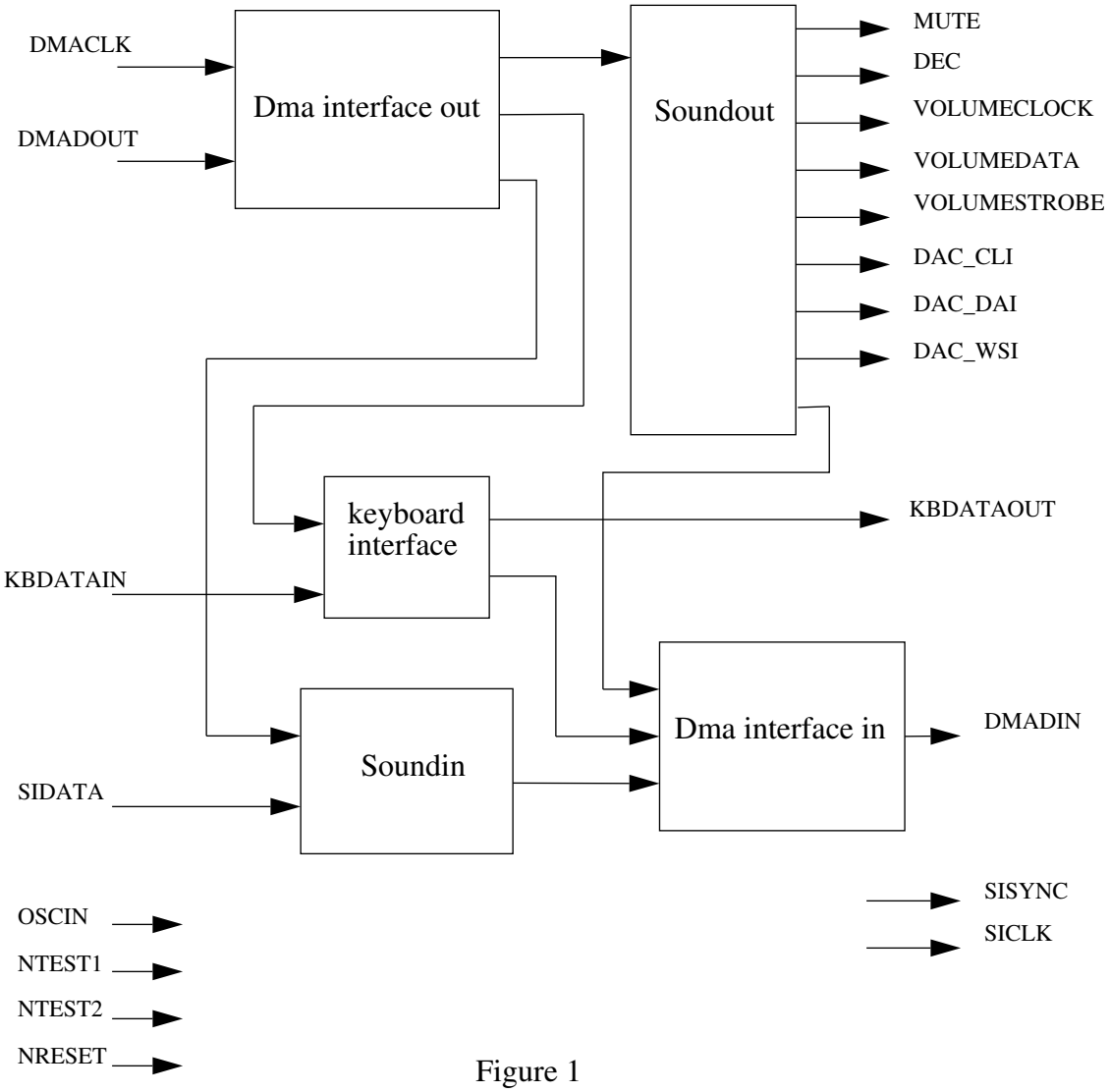


Figure 1

The asic utilizes five different serial communication protocols. One for communicating with the host, one with the keyboard and mouse, one with the DACs, one with the attenuator, and one with the Codec. The communication with the DACs and the attenuator uses also two direct (non serial) control signals: MUTE and DEC.

2.0 Pin Description

The pin assignment is described in figure 2 and the following table.

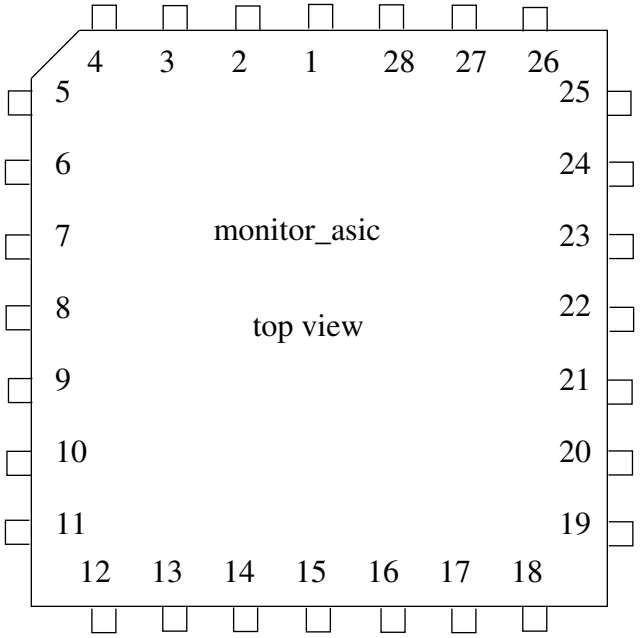


Figure 2

Pin #	Type	Pin name	Description
1	input	SIDATA	Sound in data
2	output	SISYNC	Sound in sync
3	input	NRESET	Hardware reset, active low.
4	power	IVDD2	+ 5 V +/- 10%
5	power	OVSS1	ground
6	input	DMACLK	5 MHz interface clock
7	input	DMADOUT	5 Mbit/sec data interface

Pin #	Type	Pin name	Description
8	output	DMADIN	5 Mbit/sec data interface
9	output	KBDATAOUT	async serial output
10	input	KBDATAIN	async serial input
11	power	IVSS1	ground
12	power	OVDD1	+ 5 V +/- 10%
13	output	DEC	Deemphasis control
14	output	DAC_DAI	DAC data
15	output	DAC_CLI	DAC clock
16	output	DAC_WSI	DAC word select
17	input	OSCIN	11.2896 MHz
18	power	IVDD1	+ 5 V +/- 10%
19	power	OVSS2	ground
20	output	MUTE	speaker enable
21	output	VOLUMESTROBE	attenuator strobe
22	output	VOLUMEDATA	attenuator data
23	output	VOLUMECLOCK	attenuator clock
24	input	NTEST1	selects test clocks when low
25	power	IVSS2	ground
26	power	OVDD2	+ 5 V +/- 10%
27	input	NTEST2	disables the outputs when low
28	output	SICLK	128.205 KHz

3.0 Logic Levels

NRESET, NTEST1, NTEST2, DMACLK are CMOS Schmitt Trigger inputs. KBDATAIN is a TTL Schmitt trigger input. DMADOUT is a CMOS input. SIDATA and OSCIN are TTL inputs. All inputs have pull-up resistors. All the outputs drive rail to rail. All outputs have tri-state drivers. They are in the high impedance state when the NTEST2 is low, and in the enabled state when NTEST2 is high. DMADIN can source or sink 24 mA. All other outputs can source or sink 4 mA.

Electrical characteristics extracted from the Motorola High Density CMOS Array Preliminary Design Manual and Motorola High Density CMOS Array Design Manual are summarized in the following tables:

3.1 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Valus	Unit
VDD	DC Supply Voltage	- 0.5 to 7.0	V
Vin	DC Input Voltage	- 1.5 to VDD + 1.5	V
Vout	DC Output Voltage	- 0.5 to VDD + 0.5	V
I	DC Current Drain per Pin, Any Single Input or Output	25	mA
I	DC Current Drain per Pin, Any PAralleled Outputs	50	mA
I	DC Current Drain VDD and VSS Pins	75	mA
Tstg	Storage Temperature	- 0.65 to + 150	^{o}C
TL	Lead Temperature (10 seconds soldering)	300	^{o}C

Note: Maximum ratings are those values beyond which damage to the device may occur.

3.2 RECOMMENDED OPERATING CONDITIONS (To guarantee functionality)

Symbol	Parameter	Min	Max	Unit
VDD	DC Supply Voltage	4.5	5.5	V
Vin. Vout	Input Voltage, Output Voltage	0.0	VDD	V
TA	Commercial Operating Temperature	0	+ 70	^{o}C

3.3 ELECTRICAL CHARACTERISTICS (Voltages Referenced to VSS)

Sym	Parameter	Test condition	VDD	25 °C	0 to 70 °C	Unit
VIH	Minimum High-Level Input Voltage, CMOS Input	Vout = 0.1 V or VDD - 0.1 V; |Iout| , 20 µA	4.5	2.34	3.15	V
			5.5	2.97	3.85	
	Minimum High-Level Input Voltage, TTL Input	Vout = 0.1 V or VDD - 0.1 V; |Iout| , 20 µA	4.5	1.65	2.0	
			5.5	1.94	2.0	
	Minimum High-Level Input Voltage Schmitt Trigger CMOS Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	3.25	3.1	
			5.5	3.9	4.0	
	Minimum High-Level Input Voltage Schmitt Trigger TTL Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	2.0	2.0	
			5.5	2.0	2.0	
VIL	Maximum Low-Level Input Voltage, CMOS Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	1.97	1.35	V
			5.5	2.47	1.65	
	Maximum Low-Level Input Voltage, TTL Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	1.62	0.8	
			5.5	2.85	0.8	
	Maximum Low-Level Input Voltage, Schmitt Trigger CMOS Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	1.9	1.4	
			5.5	2.4	2.2	
	Maximum Low-Level Input Voltage, Schmitt Trigger TTL Input	Vout = 0.1 V or VDD - 0.1 V; |Iout|, 20 µA	4.5	0.8	0.8	
			5.5	0.8	0.8	

ELECTRICAL CHARACTERISTICS [Continued]

IOL	Minimum Low-Level Output Current	VOL = 0.4 V 4 mA Output Types	4.5	7.4	5.8	mA
IOH	Minimum High-Level Output Current	VOH = 3.7 V 4 mA Output Types	4.5	- 7.6	- 6.3	mA
VOH (*)	Minimum High-Level Output Voltage	Vin = VIH o r VIL; IOH = -20 µA	4.5	4.5	4.4	V
			5.5	5.4	5.4	
		Vin = VIH or VIL; IOH = -4ma, -24mA	4.5	3.7	3.7	
VOL (*)	Maximum Low_Level Output Voltage	Vin = VIH or VIL; IOL = 20µA	4.5	0.001	0.1	V
			5.5	0.001	0.1	
		Vin = VIH or VIL; IOL = 4mA, 24mA	4.5	0.4	0.4	

(*) From Motorola High Density CMOS Array Preliminary Design Manual.

4.0 Host/Asic Communication Protocol.

The asic communicates with the DMA controller inside the host by means of a three-wire serial channel. One wire is the clock (DMACLK) generated by the DMA controller. The others are two unidirectional serial lines, DMADOUT and DMADIN. DMADOUT is used to send information from the DMA controller to the asic, DMADIN to send information from the asic to the DMA controller. The protocol is similar for both directions. Information is transmitted in packets. Each packet consists of a start bit, an eight-bit control field, an optional 32-bit data field and two stop bits. The control field indicates the type of packet and whether the data field is present. Packets are sent to the asic in the format given in figure 3. If the two most significant Bits of the control field are both 1, then the data field is present. If they are both 0, it is not.

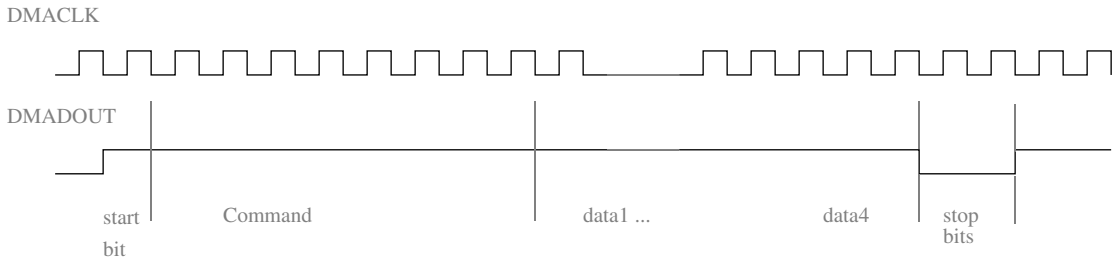


Figure 3

The asic samples the data line at the rising edge of DMACLK. The data must be valid at least 10 ns before the rising edge of DMACLK and must be held valid for at least 10 ns after the rising edge of DMACLK.

Packets are sent by the asic in the format given in figure 4.

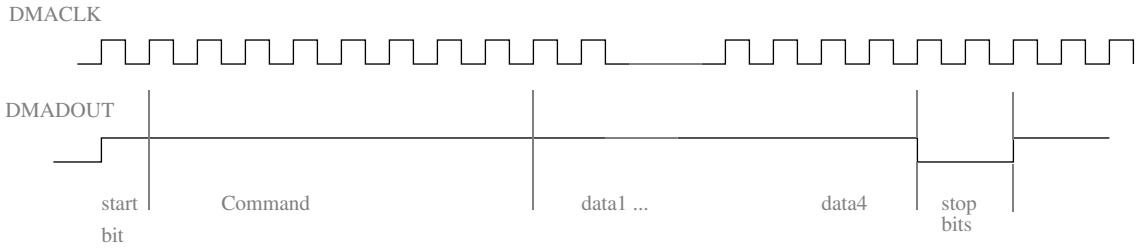


Figure 4

The asic will change the data line DMADIN after the rising edge of DMACLK.

4.1 Commands to the Asic

Commands to the asic can have two forms: Commands only and commands and data.

4.1.1 Commands Only

4.1.1.1 Commands for Sound Out

msb lsb

00 xx0 111	disable sound out
00 xx1 111	enable sound out
00 x0x 111	normal
00 x1x 111	double sample
00 0xx 111	double sample by repetition
00 1xx 111	double sample by zero filling

4.1.1.2 Commands for Sound In

msb lsb

00 xx0 011	disable soundin
00 xx1 011	enable soundin

4.1.2 Commands and Data

These commands contain a commannd field and a data field. The types of commands and data recognizeable by the asic are the following:

Command	Data1	Data2	Data3	Data4	Function
FF	FF	FF	FF	FF	Software reset
C7	left msb	left lsb	right msb	right lsb	sound packet out
C5	kb user hi	kb user lo			keyb user regiters
C4	ctrl_outputs				control outputs
C2	volume				volume

The FF command followed by FF FF FF FF data causes the reset of the asic. When the reset packet is received with a bad stop bit (i.e. stop bit = 1), the gate array will reset after a zero bit is received. Another way of looking at the reset packet is that it is a string of one bits that are longer than the longest possible packet (43 bits). The string of ones can be of any length greater than 43 bits. The gate array will generate an internal reset pulse after receiving the first zero bit. The reset pulse keeps the gate array in reset state for four clock cycles. Because of this there is a latency of 0.8 microseconds after the reset is received, before the gate array can start understanding new packets. The DMA controller must respect this latency requirement and not transmit a new packet within the first four clock cycles after giving a reset command. The same rule is valid also for the hardware reset.

Five bits of the ‘ctrl_outputs’ byte correspond to five output pins as described below:

ctrl_outputs (7, 6, 5)		unused bits
ctrl_outputs(4)	MUTE	speaker enable
ctrl_outputs(3)	DEC	deemphasis control
ctrl_outputs(2)	VOLUMECLOCK	volume clock
ctrl_outputs(1)	VOLUMEDATA	volume data
ctrl_outputs(0)	VOLUMESTROBE	volume strobe

These pins are high if the corresponding ‘ctrl_outputs’ bits are high and low if the corresponding ‘ctrl_outputs’ bits are low.

The meaning of the bits in the volume byte is the following:

Function	**bit1**	**bit0**
0 db	0	0
2 db	0	1
4 db	1	0
6 db	1	1

	bit5	**bit4**	**bit3**	**bit2**
0db	0	0	0	0
8db	0	0	0	1
16db	0	0	1	0
24db	0	0	1	1
32db	0	1	0	0
40db	0	1	0	1
48db	0	1	1	0
56db	0	1	1	1
64db	1	0	0	0
72db	1	0	0	1
infinity	1	0	1	0

	bit7	**bit6**
no change	0	0
left	0	1
right	1	0
left + right	1	1

The meaning of the bits of ‘kb user hi’ and ‘kb user lo’ is the following:

kb user hi

msb							lsb
R2	R1	R0	R/W	X	X	X	A0

kb user lo

msb							lsb
D7	D6	D5	D4	D3	D2	D1	D0

'Kb user hi' represents a command to be sent to a device (keyboard or mouse). A0 is the address of the device (0 = keyboard; 1 = mouse). R/W determines whether a read or a write operation to the device is requested (0 = write; 1 = read). If the host requires a write operation (R/W = 0) then it must supply also the second byte ('kb user lo'), which contains the data to be written. R2, R1, R0 are register selects within the device.

4.2 Commands from the Asic

Command	Data1	Data2	Data3	Data4	Function
07	no data				sound out request
0F	no data				sound out underrun
C7	sound1	sound2	sound3	sound4	sound in data
C6	paddr	rev #	data msb	data lsb	kb receive data

rev # =0000 0000 for the old asic
0000 0001 for the new asic

Command 07 and 0F can happen only when sound out is enabled. 07 requests new sound packets to the DMA controller. 0F informs the host that a sound packet has either not arrived or not arrived on time. Command C7 can happen only when sound in is enabled.

'Sound1' is the oldest byte of sound in data; 'sound4' is the newest. 'Data lsb' is the first byte received from a device and corresponds to a keycode; 'data msb' is the second. If the device is a keyboard, data lsb corresponds to a keycode and 'data-msb' corresponds to a metakey code.

The bits of ‘paddr’ have the following meaning:

messages caused by an asic originated poll event

0 x 0 d a a a a	d = 0 not master keyboard;	d = 1 master keyboard
0 d 0 x a a a a	d = 0 normal event;	d = 1 no response error (device unplugged)

aaaa = address of device responding
0000 = keyboard
0001 = mouse

responses to a user originated event

0 x 1 d x x x x	d = 0 valid data	d = 1 invalid data
0 d 1 x x x x x	d = 0 normal event	d = 1 no response error (device unplugged)

5.0 Sound Out Protocol

The sound out signals DAC_CLI, DAC_WSI, DAC_DAI generated by the asic are compatible with the requirements of the Philips SAA7320 Stereo CMOS DAC for compact disk digital audio system. The timing relationship of these signals is illustrated in figure 5.

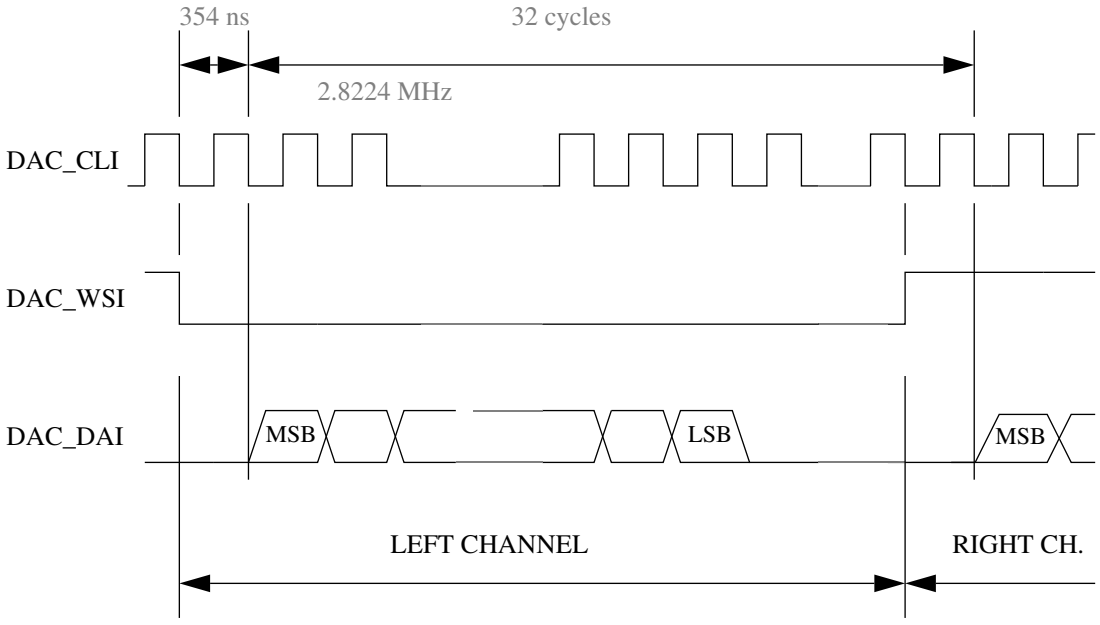


Figure 5

DAC_CLI is a clock signal. DAC_DAI carries data in the form of two 16-bit words, to be applied to the DACs, for the left and right sound channels. DAC_WSI selects left or right channel.

6.0 Attenuator Control Protocol

VOLUMECLOCK, VOLUMEDATA, VOLUMESTROBE are used to control the attenuator. These signals are compatible with the requirements of the attenuator gate array. VOLUMEDATA is used to send volume information in a serial way to the attenuator, MSB first. VOLUMECLOCK is used to shift VOLUMEDATA into the attenuator, and VOLUMESTROBE to transfer the data into the latches of left and right channels of the attenuator as defined by bits 6 and 7 of VOLUMEDATA. In the previous version of the asic the task of generating the proper sequential patterns for these signals was handled by software; every transition required a new packet. In this version volume information can be sent from the host to the asic either as in the old version using the 'ctrl_outputs' byte, or by using the 'volume' byte. If it is sent as a 'volume' byte, the software needs to only send one packet, while the asic generates the right sequence of the three signals. The timing parameters when the signals are generated by the asic are given in figure 6.

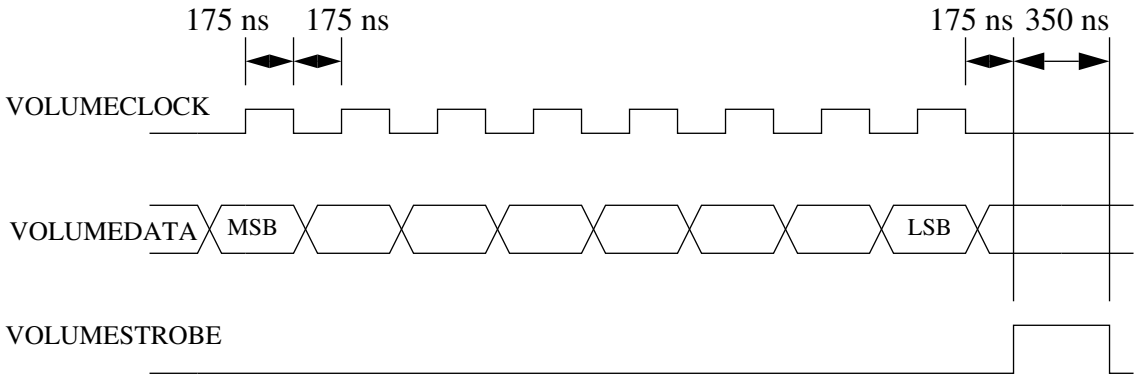


Figure 6

VOLUMECLOCK, VOLUMEDATA and VOLUMESTROBE are connected inside the asic to the output of a multiplexer, which selects either three signals provided as 'ctrl_outputs' or three signals generated internally, from information contained in the 'volume' byte. After reset, the multiplexer selects the three ctrl_outputs signals, and the asic appears the same as the old version. If the asic receives a C2 vv xx xx xx packet, the multiplexer will select

the three signals generated internally, and will stay in that position until it receives a hardware or software reset.

7.0 Sound In Protocol

When sound in is enabled, the asic generates SISYNC and SICLK; the Codec provides SIDATA.These signals are compatible with the Motorola PCM Codec/Filter Mono-Circuit, MC145503. A timing relationship of these signals is given in figures 7 and 8.

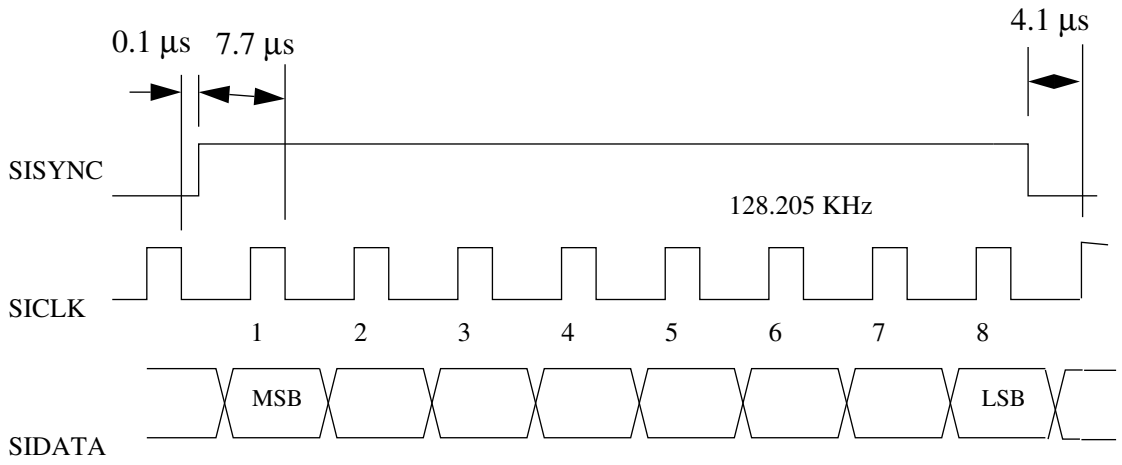


Figure 7

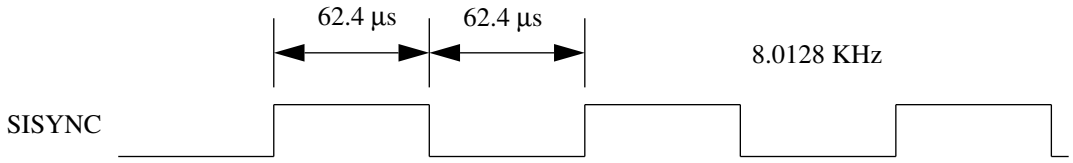


Figure 8

In normal mode of operation the frequency of SICLK is obtained by dividing the frequency of DMACLK by 39. When NTEST1 is low it is obtained by the frequency of DMACLK divided by 3. The duty cycle of SICLK is 1:3 when in test mode, and 6:13 when in normal mode.

8.0 Keyboard Interface Protocol

This version of the asic can control only one mouse and one keyboard, which by default is also the master keyboard. The addresses of the keyboard (0000) and mouse (0001) are hard-wired.

The communication between the asic and the devices (keyboard or mouse) takes place by means of two asynchronous serial lines, one to transmit data to the devices, the other to receive data from the devices. Serial data rate is 18,958 +/- 2% bits/sec. A digital phase locked loop using an 8X clock, KBCLK, is used. KBCLK has a frequency of 151.515 KHZ. It is obtained by dividing DMACLK by 33. Data is sent in packets. The format of a packet is illustrated in figure 9.

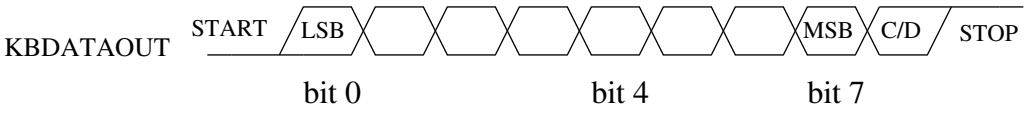


Figure 9

A packet contains one start bit, eight data bits, a command/data indicator bit, and a stop bit. Of the data bits the lsb is transmitted first and the msb last. Stop bits are active high. The command/data indicator bit is high to indicate that the byte is a command, low to indicate that it is a data. Data bits are high if they represent a one and low if they represent a zero.

The asic acts as a master and sends commands to the devices. Commands are requests for write or read operations. If a write is requested, bit 4 of the command byte is zero; if a read is requested it is one. When a read is requested, the asic, after issuing the read command waits for the answer from the devices. When a write is requested, the asic provides the data to be written with the next packet. The next packet will have the command/data indicator bit set to zero. Below is a description of the bits in a command and data byte. A3 to A0 contain the address of the devices. A3 through A1 are always zero. A0 is 0 for the keyboard and 1 for the mouse. R2, R1 and R0 select registers within a device.

Command byte (C/D = 1)

msb							lsb
R2	R1	R0	R/W	A3	A2	A1	A0

Data byte (C/D = 0)

msb							lsb
D7	D6	D5	D4	D3	D2	D1	D0

The commands transmitted to the devices can be classified in two categories: commands generated by asic, and commands generated by the host and sent to the devices via the asic. The latter are also referred to as user commands. User commands are sent to the asic by the host as the ‘kb user hi’ and ‘kb user lo’ bytes, following a C5 DMA command. ‘Kb user hi’ corresponds to the command byte. ‘Kb user low’ corresponds to the data byte. The commands originated by the asic are poll commands, and therefore are only read (bit 4 = 1). They are the following:

msb	lsb	
0001	0000	poll keyboard
0001	0001	poll mouse

The commands generated by the host are the following:

user write commands

msb	lsb	msb	lsb	
0000	1111	0000	0000	reset
1110	1111	xxxx	aaax	set address (used by old software)
0000	aaa0	xxxx	xxcd	set keyboard status c = led2 (1=on); d = led1(1=on)

user read commands

msb	lsb	
1111	aaa0	read version of keyboard
1111	aaa1	read version of mouse
0001	aaa0	poll keyboard
0001	aaa1	poll mouse

In communicating with the devices the asic goes through a transmit/receive cycle. The cycle may be utilized for user commands or for poll commands originated by the asic. User commands have priority over asic originated poll commands. The asic alternates its poll commands in keyboard polls and mouse polls. The transmit/receive cycle consists of 41 steps, as illustrated in figure 10. Each step has the duration of one bit being transmitted or received through the serial interface. At step 19 the asic issues a start bit. From steps 20 to step 27 it sends data bits. At step 28 it issues the command/data bit. At steps 29 and 30, two stop bits.

If the command bit is 1 and data bit 4 is zero, then at step 31 it issues a start bit to start transmitting the second packet, which contains data to be written into the device. From steps 32 to 39 it sends data bits. At step 40 it transmits the command/data bit as a zero, to indicate that it is a data byte. At step 41 it issues the a stop bit.

If a read is requested, from steps 29 to the next step 18 the asic will listen to the response from the devices. The devices are required to answer within 2 bits of the stop bit of the read request, with two bytes of data (command/data bit = 0), corresponding to the R2, R1 and R0 bits of the poll byte. These two bytes will be interpreted as a 16-bit quantity, with the first byte representing the LSB and the second the MSB. If no response is received within this time, the asic will inform the host that the keyboard has been disconnected. This is done through bit 6 (1 = no response) of the paddr byte, following a C6 DMA command to the host. If the keyboard has no new data to send it will respond with two bytes of zeros, with the data/command bit set to command (C/D = 1). The ‘keyboard interface’ requests the ‘dma interface in’ to send data to the host only in the following three cases:

1) The data coming from the devices is a response to a user command;
2) The devices do not answer on time (no response);
3) The data coming from the devices is a response to an asic originated poll command and and the command/data bit bits set to data (C/D = 0).

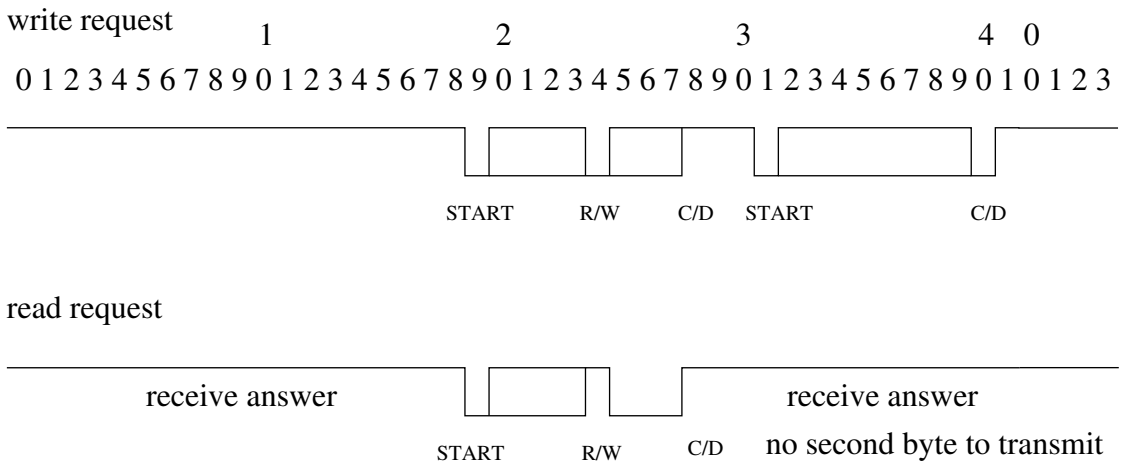


Figure 10

9.0 Monitor Asic Revision Number

The revision number of the previous version of the asic is 00000000. The revision number of this version is 00000001. The asic revision number is sent to the host together with the keyboard or mouse version number, as the second byte of data, following a request of id of keyboard or mouse by the host. The protocol is the following:

Host:	C5 F0 xx xx xx xx	read version of keyboard, or
	C5 F1 xx xx xx xx	read version of mouse.

Asic :	C6	paddr	asic_rev#	device_version	device_id.

10.0 Software Compatibility

This asic is designed to be totally compatible with the existing software (release 1.0 or earlier versions) at power up.

Because the address of the keyboard and mouse are hard-wired, software does not need to set or refresh them. Any attempt to write over them will be ignored by the asic, and the system will remain functional also with the old software which attempts to set these addresses.

At power up the asic is set to accept attenuator control information as in the old version. To send attenuator control information with the new method, software must simply send a C2 vv xx xx xx packet, where vv is the volume byte. Upon receiving of the C2 command the asic switches itself into the new mode, in which the asic generates the attenuator control signals from information contained in the ‘volume’ byte. The asic remains in the new mode until a hardware or software reset is given to it.

Since new versions of software may be running on systems with old or new version of the asic, it is important that new software interrogates the asic to find out its revision number, and make sure that a particular monitor contains the new version of the asic, before attempting to use the new features.

11.0 Test Modes

The monitor asic has two test pins: NTEST1 and NTEST2 . They are both active low. During normal operation they can be left unconnected or connected to IVSS2. If they are left unconnected they are kept high by the pull-up resistors. NTEST1 is used to bypass internal counters during the testing of the asic. NTEST2 is used to disable the output buffers and improve testability of the printed circuit board.

12.0 Design Files

The design files are kept under two directories. One is called MONITOR_ASIC_EVERY-THING, the other is called MONITOR_ASIC. The MONITOR_ASIC_EVERYTHING contains all the design files, two global simulation cases - one at full speed and one at Motorola's tester speed -, plus fragments of simulation cases used during the development phase. The MONITOR_ASIC directory is the directory that has been given to Motorola to manufacture the part. This directory is a copy of the MONITOR_ASIC_EVERYTHING, without the fragments of simulation cases.

The name of the design is MON. MON is also the name of the outer block. Inner blocks are DMAINT_OUT, DMAINT_IN, SOUNDOUT, KBINTERFACE, SOUNDIN, and CLOCKS.
The global simulation case at full speed utilizes the following two files:
MON_FULSPEED_DRIVER and MON_FULSPEED_STIM.MISL.
The global simulation case at tester speed utilizes the following two files:
MON_DRIVER and MON_STIM.MISL

The true capacitance file provided by Motorola and based on the actual routing is mon.actcap and it is located in the monitor_asic/mon/vita directory.

13.0 Design Methodology

The design has been done using a Mentor workstation and the revision 3.32 of Motorola software for High Density CMOS Array DVM. The following packages of the 3.32 DVM have been utilized:

ASIC_NETED
ASIC_EXPAND
MENTOR2EDIF
DECAL
INSERT_DELAYS
ASIC_QUICKSIM
SIMCAP
SIMCHECK.

Schematic capture was done using ASIC_NETED. Simulation using ASIC_QUICKSIM. Instructions on how to use the Motorola package are provided by Motorola in the Motorola High Density CMOS Array DVM Mentor Graphics User's Guide Supplement, software release 3.32. Stimuli patterns were built using the Mentor Interactive Stimulus Language (MISL) provided by Mentor.
Each simulation case uses a MISL file, and two DO files. One DO file is called START. The other has a name that reminds of the simulation case that it refers to, and ends with the word _DRIVER. The START file is executed once for each simulation case, before the simulation time has advanced, to save the simulation state at time zero for further use.

To execute a simulation case, once in ASIC_QUICKSIM, one needs to first execute the START file by typing:

DO START.

Then execute the driver file that refers to the particular simulation case to be simulated. This can be done by typing:

DO <casename_DRIVER>.

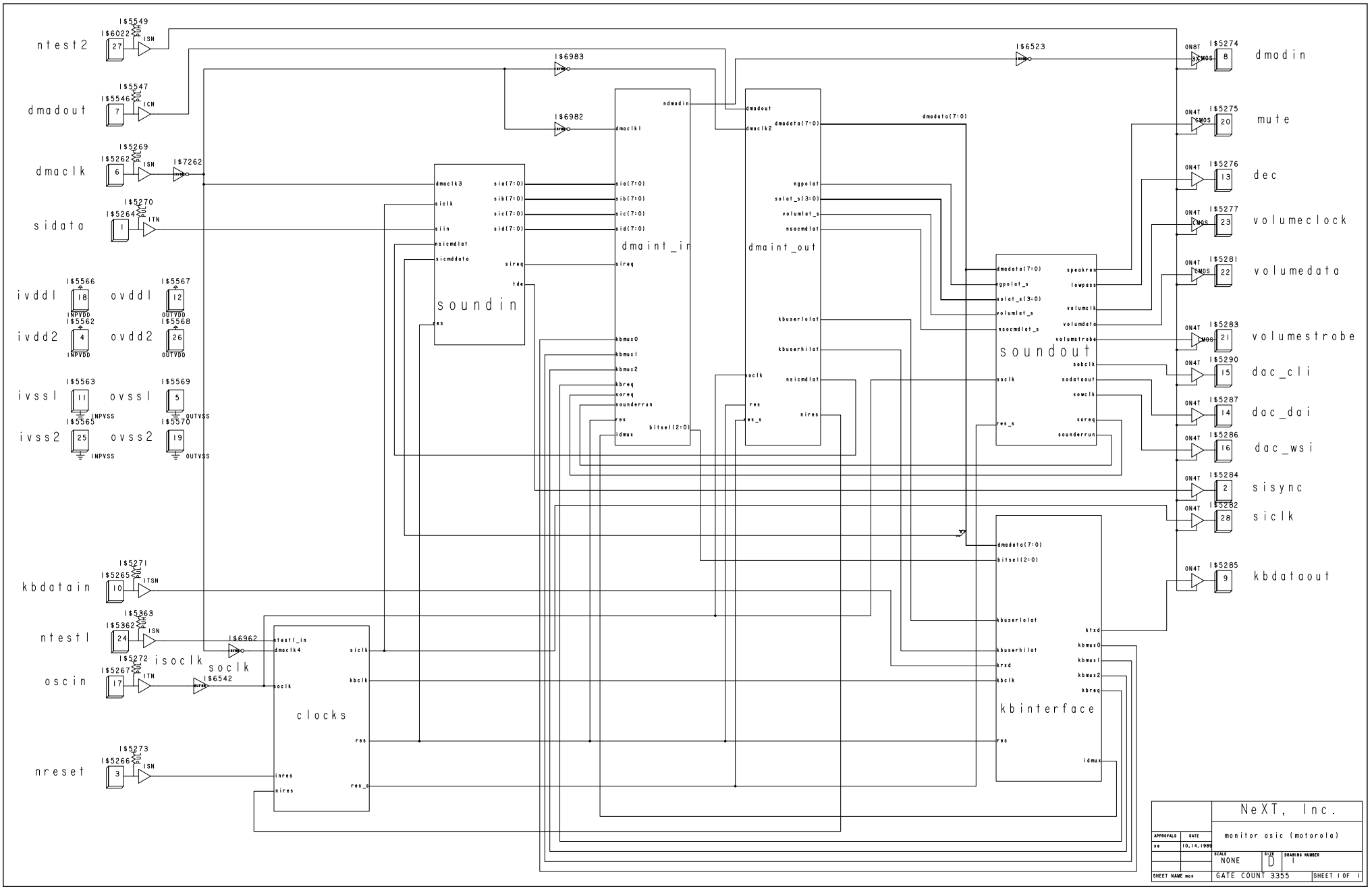
Execution of this file will restore the state at time zero, invoke the MISL compiler to compile the MISL file, and issue the run command to the simulator.

Example 1:
do start
do mon_driver

Example 2:
do soundin3_driver.

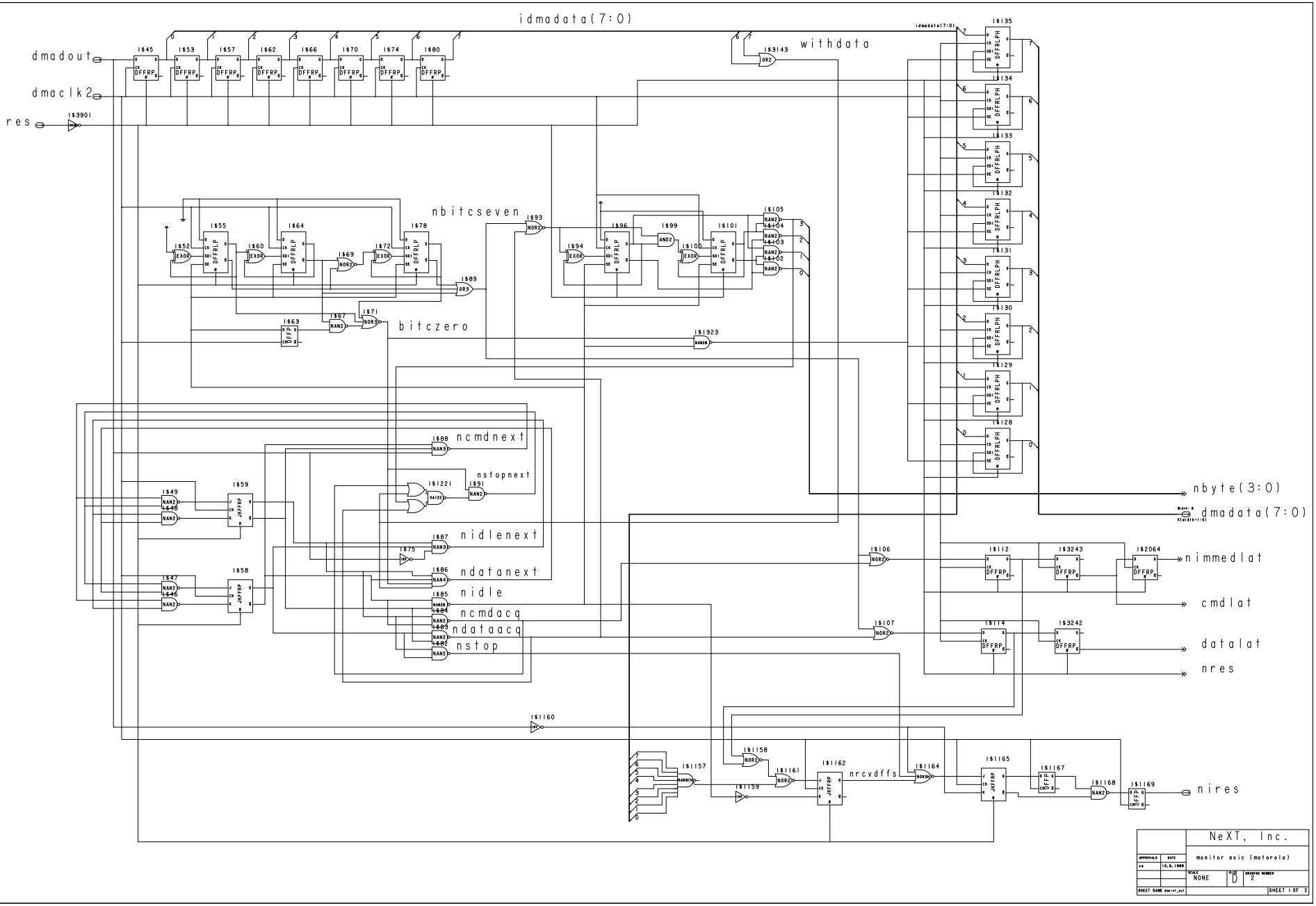
14.0 Attached Schematics

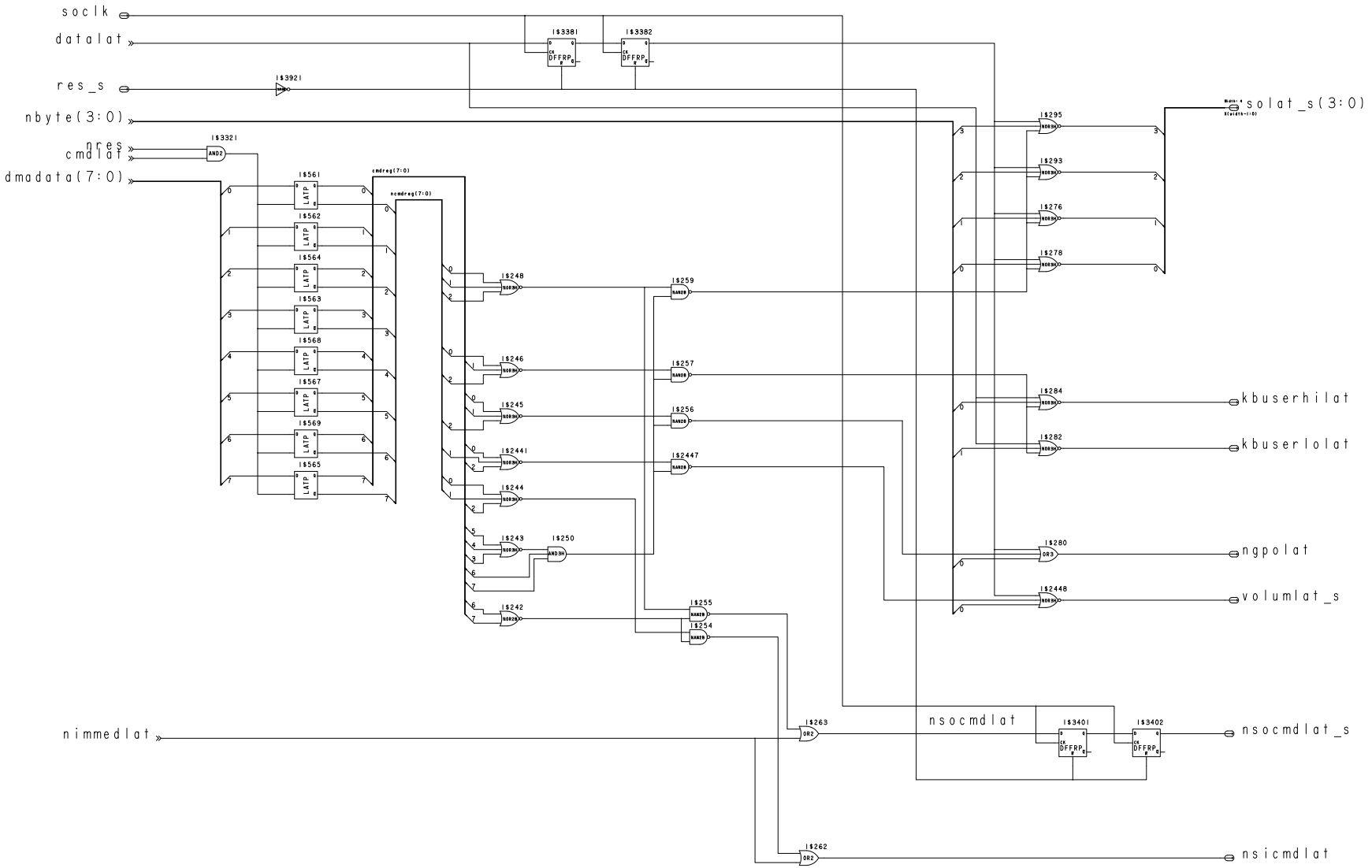
The next 12 pages list the schematics for the ASIC.

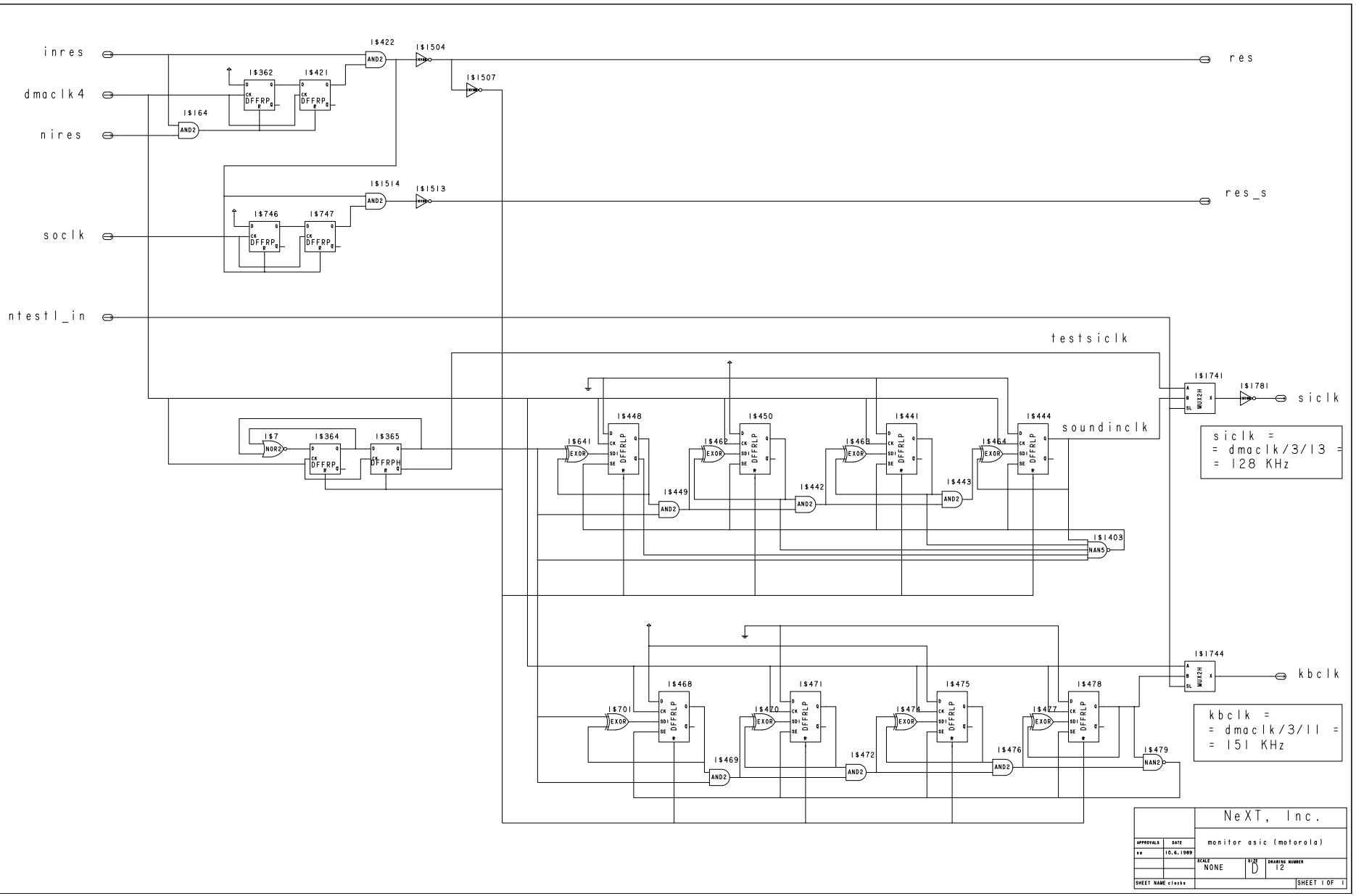


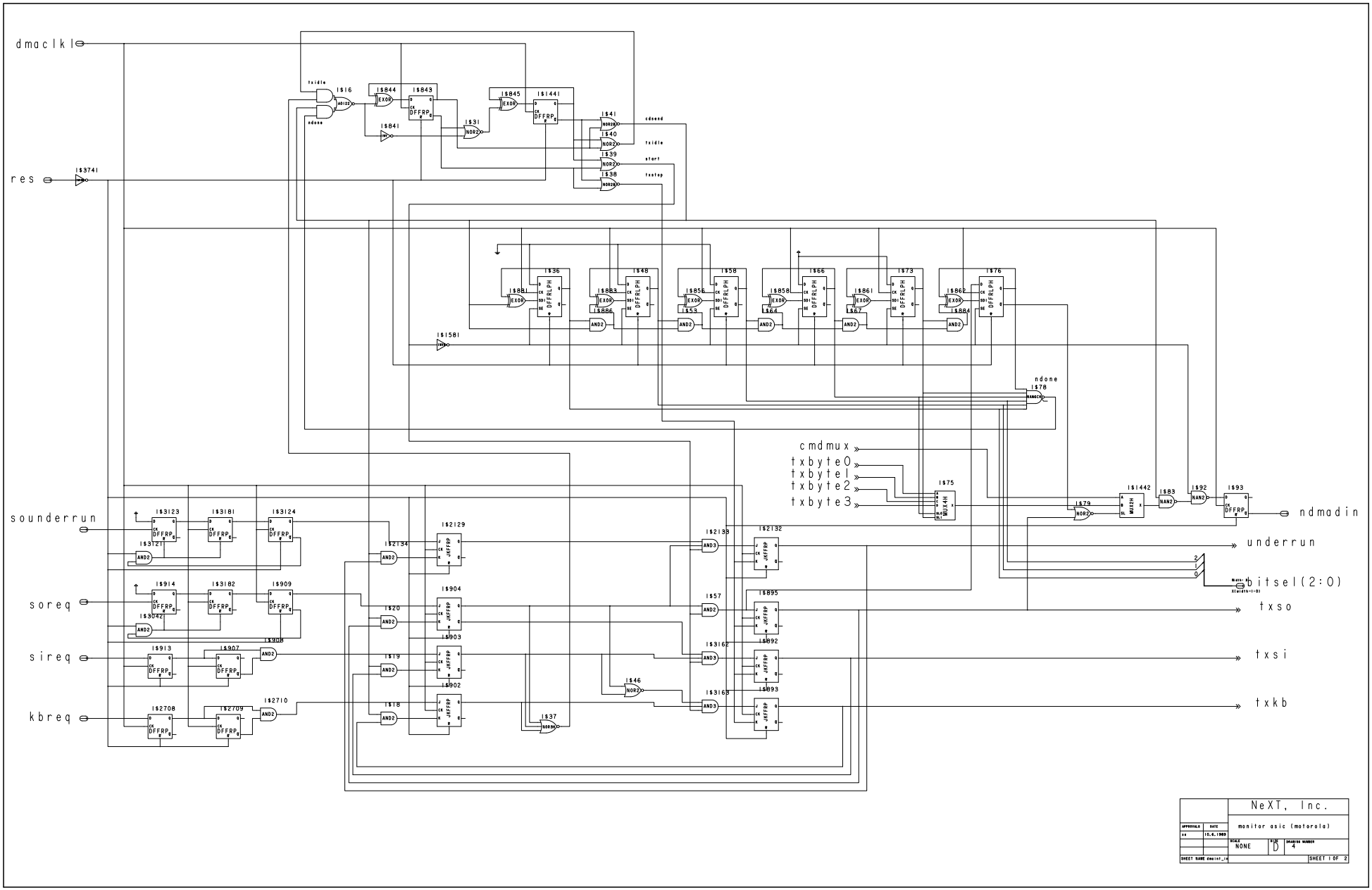
	NeXT, Inc.	
APPROVALS	DATE	monitor asic (motorola)
se	10.14.1989	
	SCALE NONE	SIZE D DRAWING NUMBER 1
SHEET NAME men	GATE COUNT 3355	SHEET 1 OF 1

	Document # **1162.00**	
	Size **A**	Sheet 23 of 34

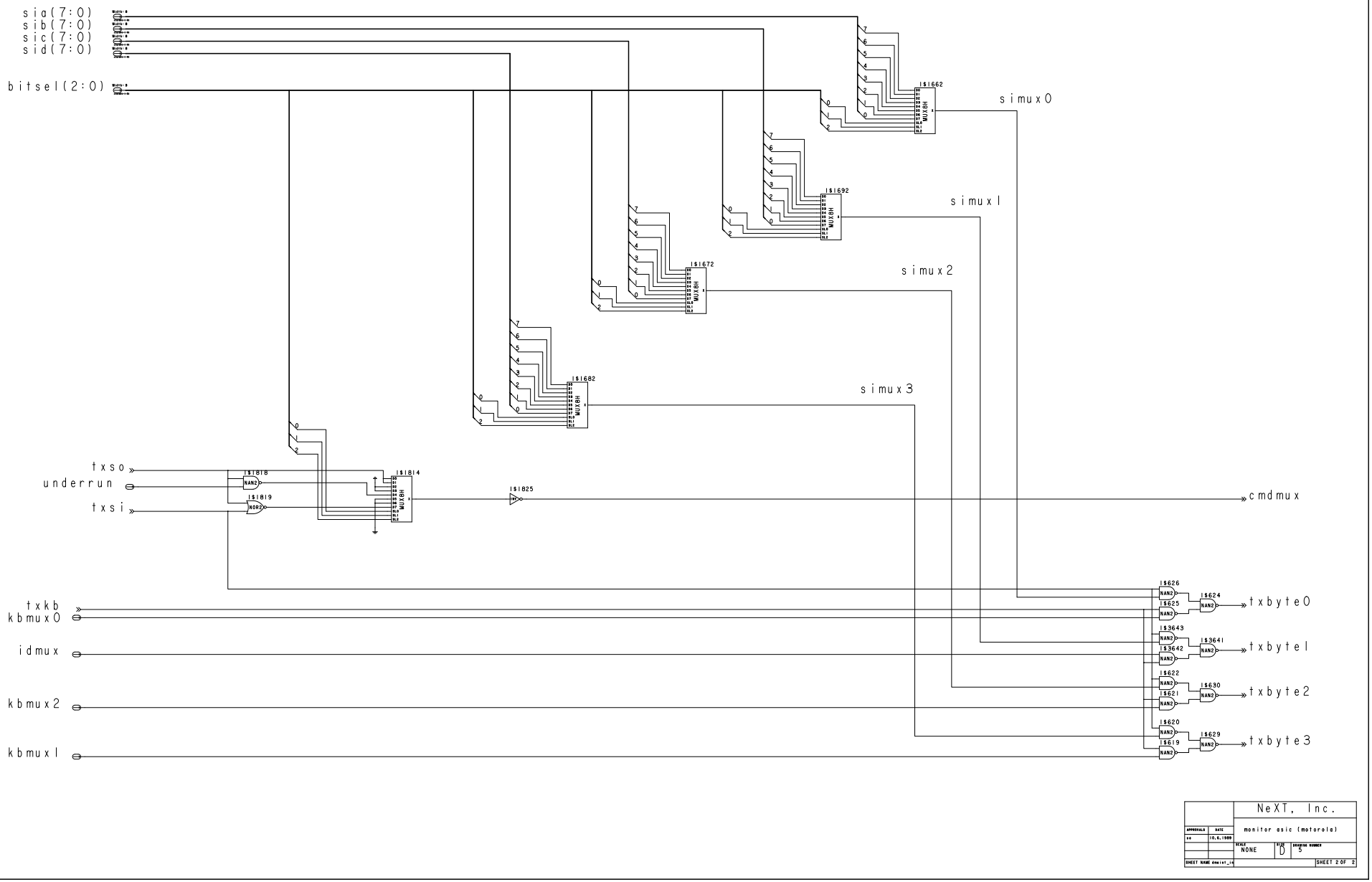

idmadata(7:0)
dmadout
dmaclk2
res
withdata
nbitcseven
bitczero
ncmdnext
nstopnext
nidlenext
ndatanext
nidle
ncmdacq
ndataacq
nstop
nrcvdffs
nbyte(3:0)
dmadata(7:0)
nimmedlat
cmdlat
datalat
nres
nires
NeXT, Inc.
monitor asic (motorola)
SCALE NONE
SIZE D
DRAWING NUMBER 2
SHEET 1 OF 2


soclk
datalat
res_s
nbyte(3:0)
nres
cmdlat
dmadata(7:0)
nimmedlat
I$3381
I$3382
I$3921
I$3321
I$561
I$562
I$564
I$563
I$568
I$567
I$569
I$565
cmdreg(7:0)
ncmdreg(7:0)
I$248
I$246
I$245
I$2441
I$244
I$243
I$250
I$242
I$259
I$257
I$256
I$2447
I$255
I$254
I$263
I$262
I$295
I$293
I$276
I$278
solat_s(3:0)
I$284
I$282
kbuserhilat
kbuserlolat
I$280
ngpolat
I$2448
volumlat_s
nsocmdlat
I$3401
I$3402
nsocmdlat_s
nsicmdlat
NeXT, Inc.
monitor asic (motorola)
NONE
SHEET 2 OF 2


inres
dmaclk4
nires
soclk
ntestl_in
res
res_s
testsiclk
soundinclk
siclk
kbclk
siclk =
= dmaclk/3/13 =
= 128 KHz
kbclk =
= dmaclk/3/11 =
= 151 KHz
I$422
I$1504
I$1507
I$362
I$421
I$164
I$1514
I$1513
I$746
I$747
I$7
I$364
I$365
I$448
I$450
I$441
I$444
I$641
I$462
I$468
I$464
I$449
I$442
I$443
I$1403
I$1741
I$1781
I$1744
I$468
I$471
I$475
I$478
I$701
I$470
I$474
I$477
I$469
I$472
I$476
I$479
NeXT, Inc.
monitor asic (motorola)
APPROVALS
DATE
10.6.1989
SCALE
NONE
SIZE
D
DRAWING NUMBER
12
SHEET NAME clocks
SHEET 1 OF 1



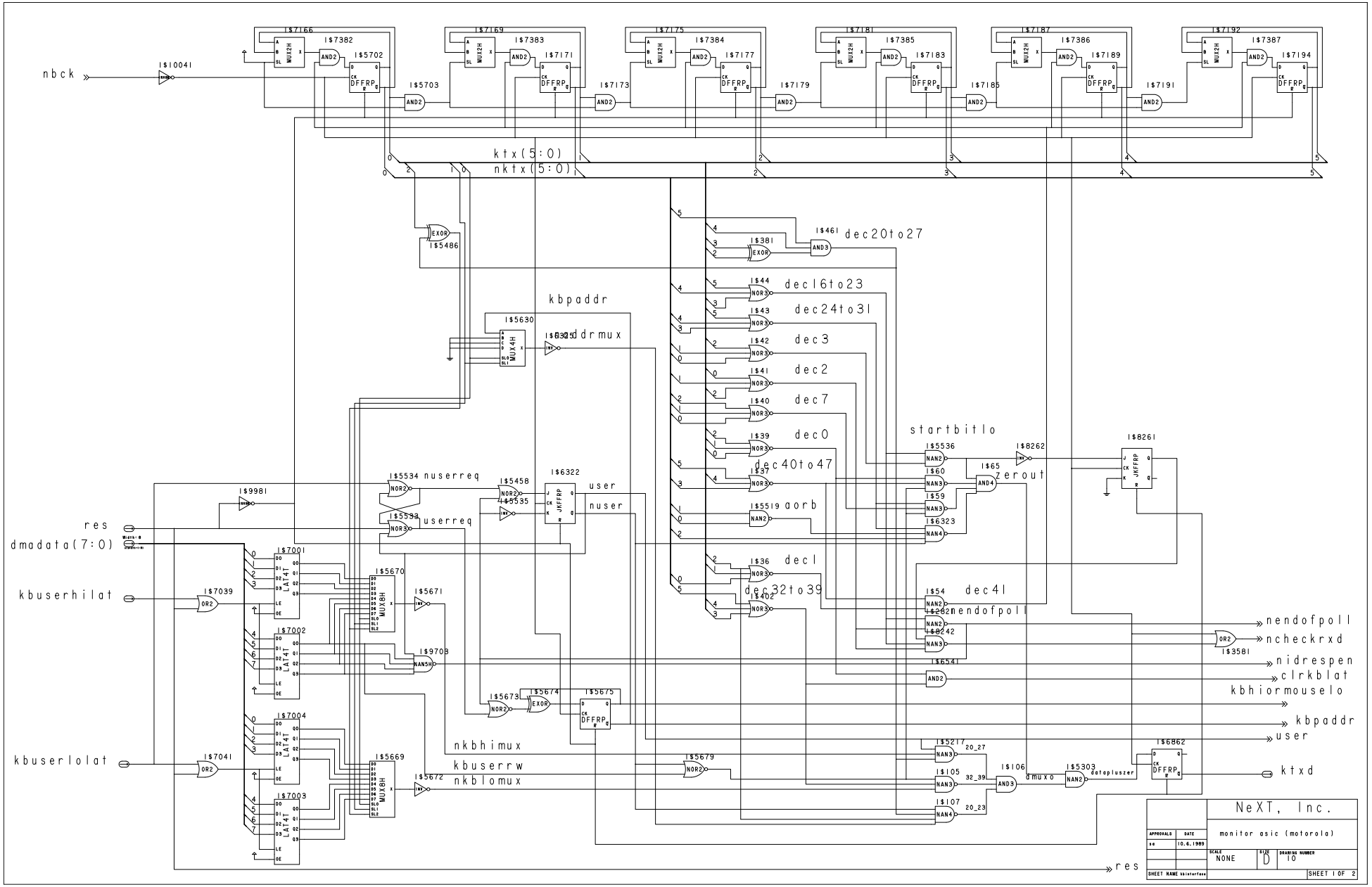
Document #	**1162.00**
Size	A
	Sheet 27 of 34



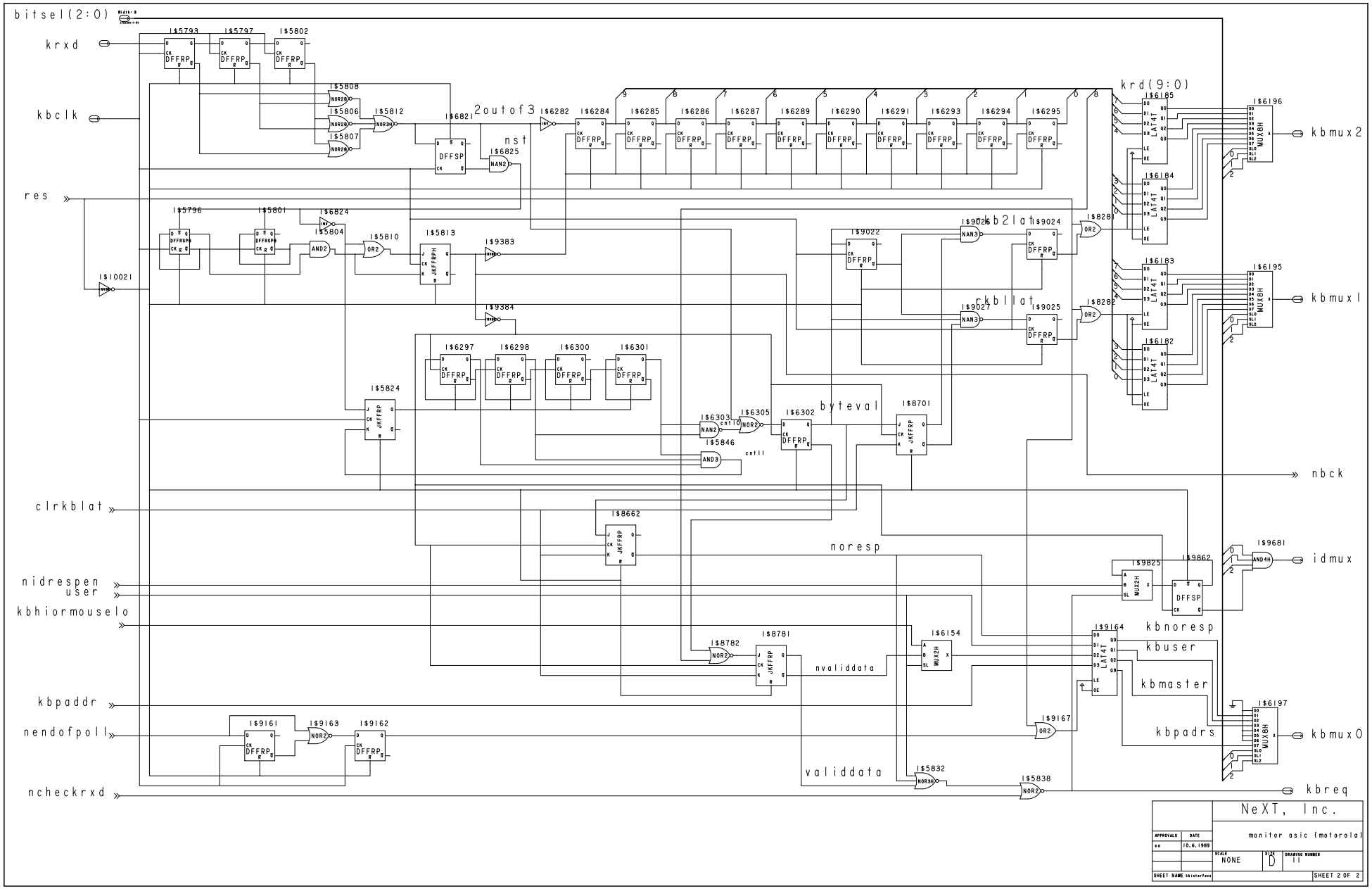
Document # **1162.00**

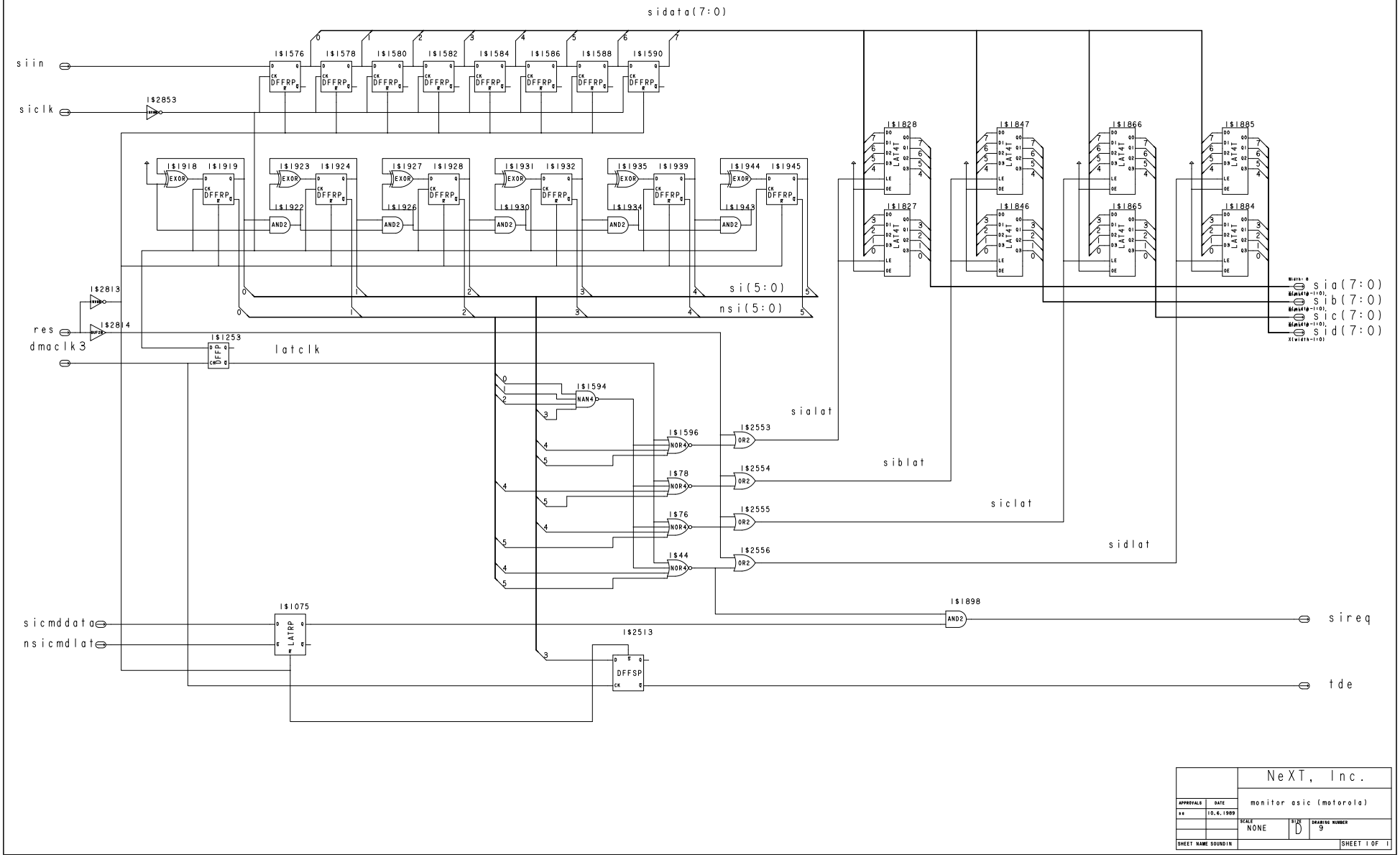
Size **A**

Sheet 28 of 34



Document #	**1162.00**
Size	A
	Sheet 29 of 34

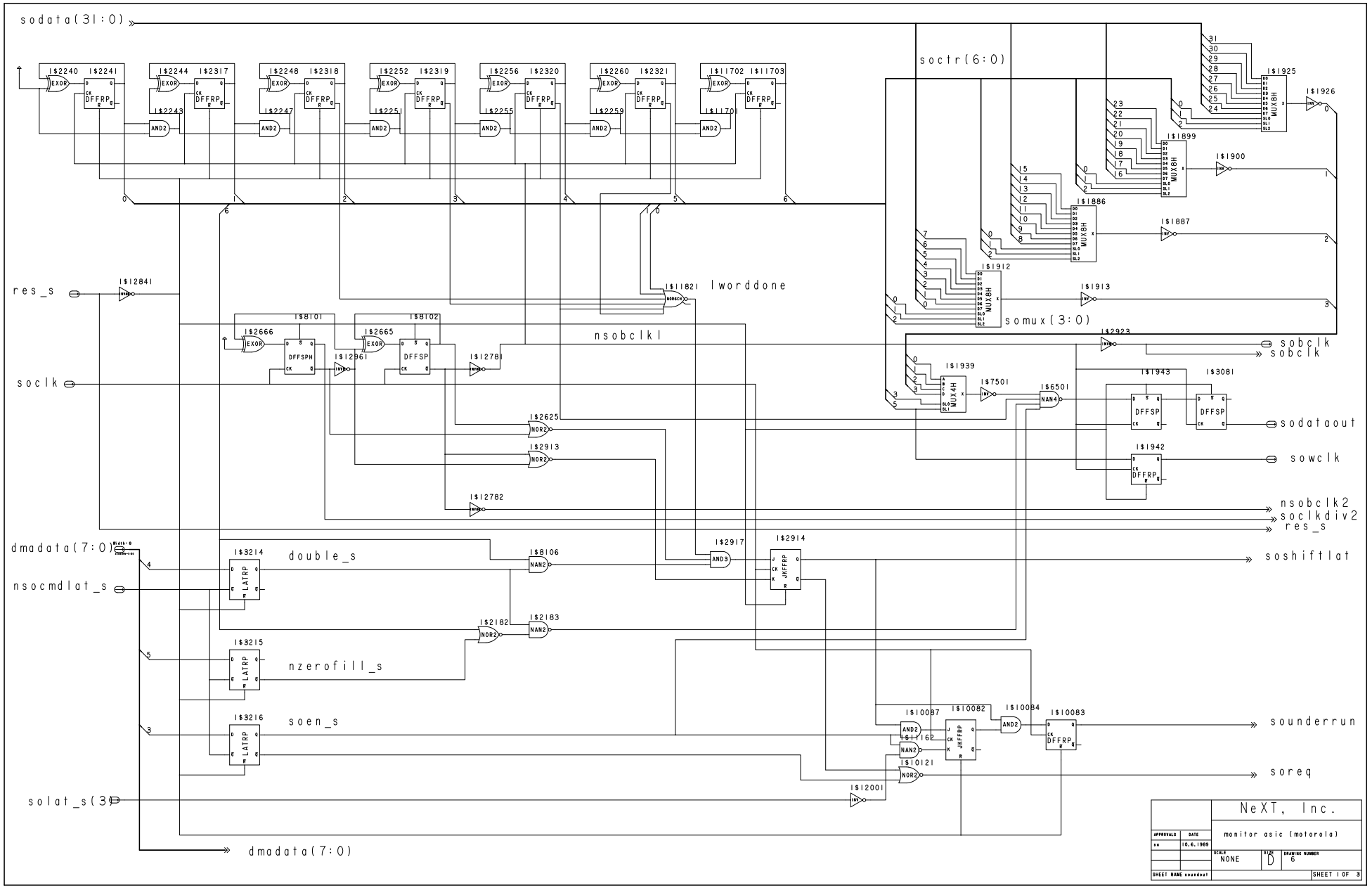

bitsel(2:0)
krxd
kbclk
res
clrkblat
nidrespen
user
kbhiormouselo
kbpaddr
nendofpoll
ncheckrxd
krd(9:0)
2outof3
nst
kb2lat
rkbllat
byteval
noresp
nvaliddata
validdata
kbmux2
kbmux1
kbmux0
nbck
idmux
kbnoresp
kbuser
kbmaster
kbpadrs
kbreq
NeXT, Inc.
monitor asic (motorola)
SCALE NONE
SIZE D
DRAWING NUMBER 11
SHEET 2 OF 2



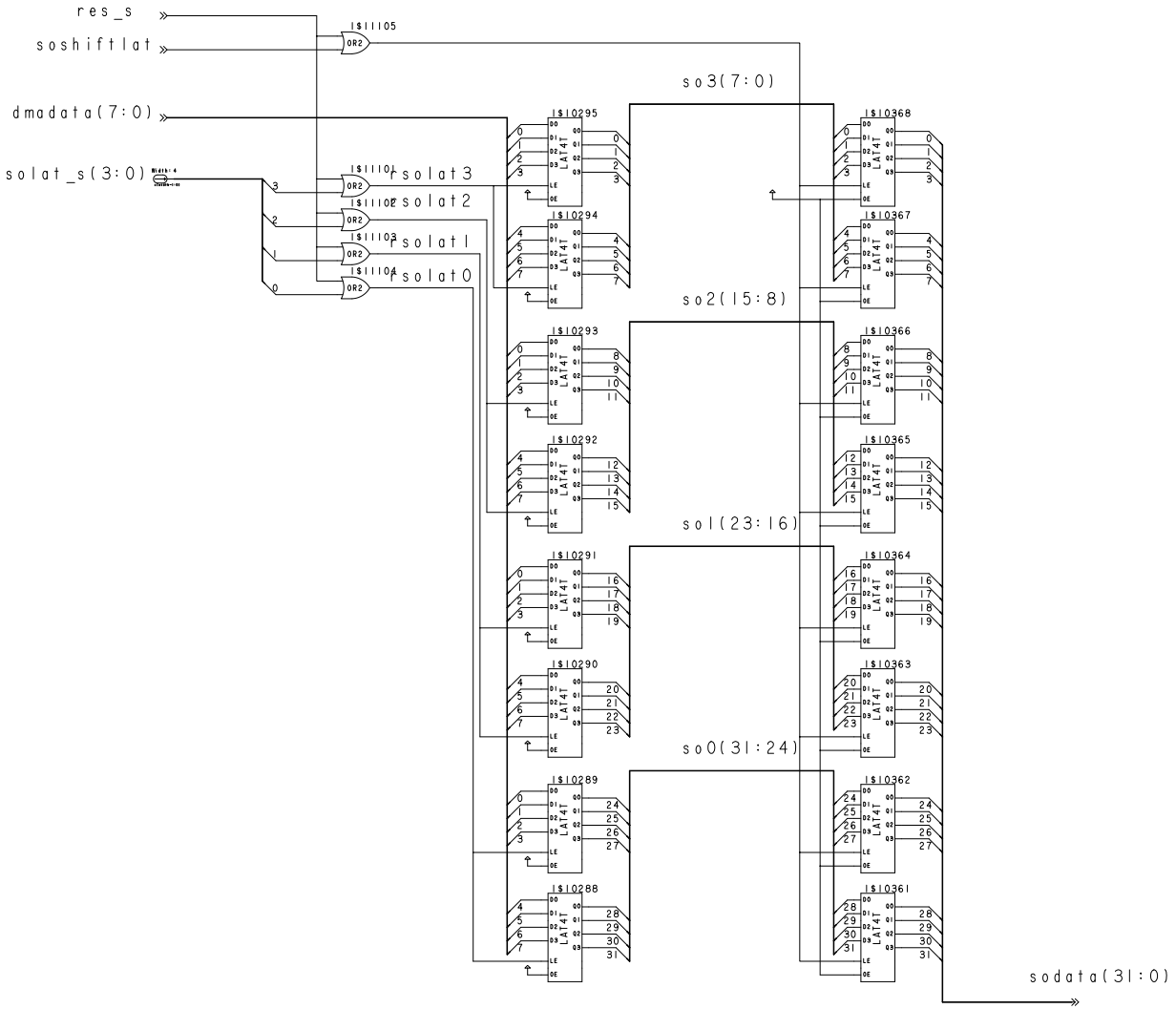
Document # **1162.00**

Size **A**

Sheet 31 of 34

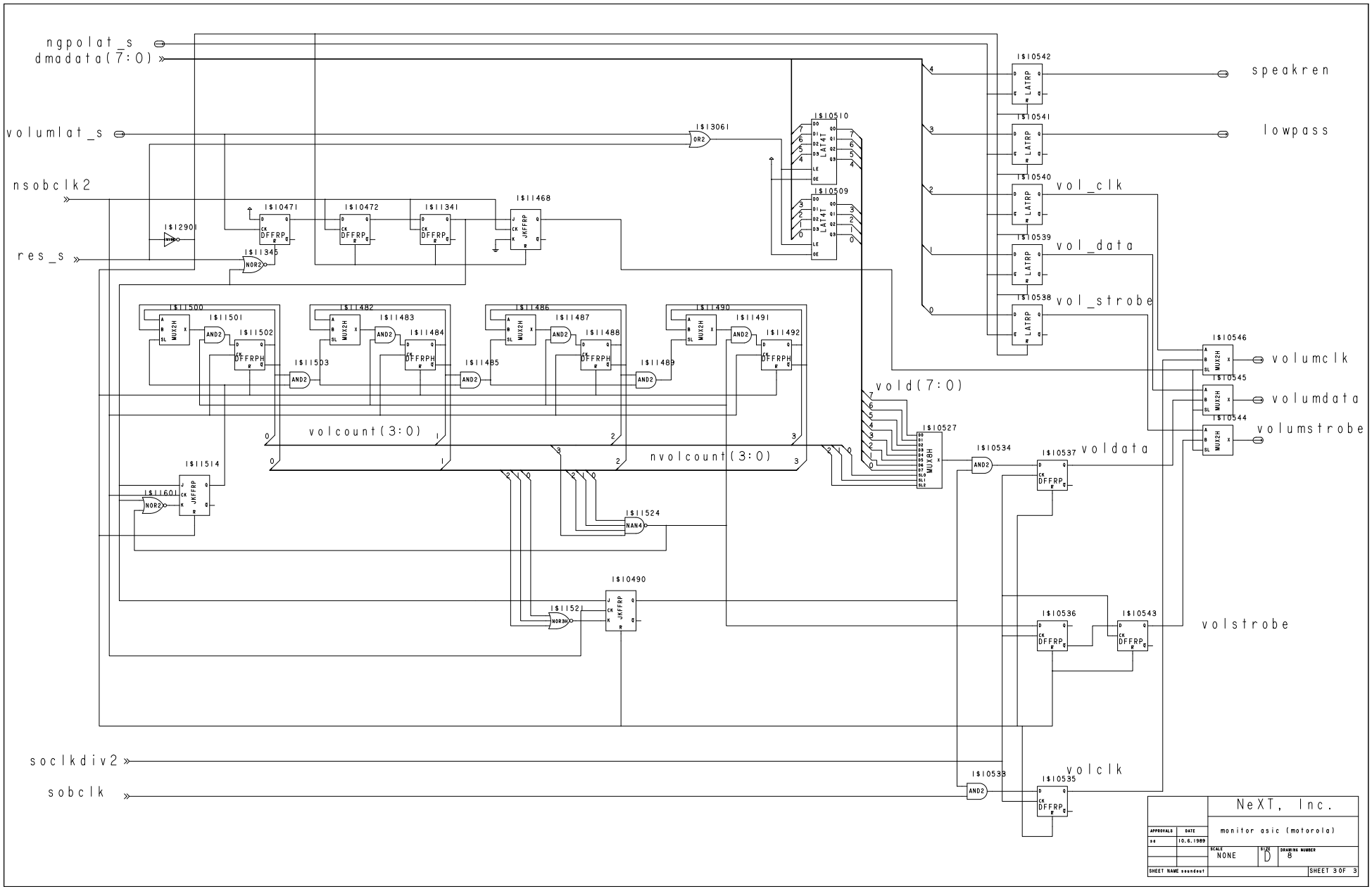


Document #	**1162.00**
Size	A
	Sheet 32 of 34



		NeXT, Inc.
APPROVALS	DATE	monitora sic (motorola)
sa	10.6.1989	
		SCALE NONE · SIZE D · DRAWING NUMBER 7
SHEET NAME soundout		SHEET 2 OF 3

NeXT™	Document # **1162.00**	
	Size **A**	Sheet 33 of 34



Document # **1162.00**

Size **A**

Sheet 34 of 34