DK4000-C167
USER MANUAL

Development Kit for PSD4000 and C167

CONTENTS

Congratulations on purchasing ST's DK4000 Development kit. The DK4000 (110 or 220 volt version) is a low cost kit for evaluating the PSD4000 series of Flash Programmable System Devices called PSDs. The DK4000 kit is extremely versatile, and can be used in several different modes. It can be used to demonstrate the PSD4000's capability of JTAG In-System Programmability (ISP). Once initial code is resident in the PSD, the program code can be updated while the MCU is running, called In-Application Programming (IAP). Also, Infineon C167CR family users can utilize the DK4000 as an evaluation platform for code development.

The DK4000 – C167 Development Board is specific to the Infineon C167CR micro-controller family. However, other proliferation boards will be available. Check the website at *www.st.com/psd* as to availability.

A COUPLE OF DEFINITIONS

In-System Programming (ISP) - A JTAG interface (IEEE 1149.1 compliant) is included on the PSD enabling the entire device to be rapidly programmed while soldered to the circuit board (Main Flash memory, Secondary Boot Flash memory, the PLD and all configuration areas). This requires no MCU participation, so the PSD can be programmed or reprogrammed anytime, anywhere, even while completely blank. The MCU is completely bypassed.

In-Application Programming (IAP) – Since two independent Flash memory arrays are included in the PSD, the MCU can execute code from one memory while erasing and programming the other. Robust product firmware updates in the field are possible over any communication channel (a few examples are CAN, Ethernet, UART, J1850) using this unique architecture. For IAP, all code is updated through the MCU.

HARDWARE

- PSD4000 Flash PSD (Programmable System Device) - see *www.st.com/psd* for data sheet. PSD4135G2 - 4Mbit Main Flash memory (512Kx8), 256Kbit Boot Flash memory (32Kx8), 64Kbit SRAM (8Kx8)

- Eval/Demo Board with C167CR or other MCU, LCD Display, JTAG and UART ports for ISP/IAP
- FlashLINK JTAG ISP Programmer (uses PC's parallel port)
- Straight through serial cable (Male-Female)
- Power Supply

SOFTWARE

To ensure you have the latest versions, check the website often.

- PSDsoft Express - Point and Click Windows programming development software. This will install to it's own directory.
  - MCU Selection by manufacturer and part number
  - Graphical definition of pin functions
  - Easy creation of memory map
  - JTAG ISP Programming.
- Downloadable demonstration software
  - Visit the development tools section of ST PSM product division website at www.st.com/psd to download the file 167_disk.zip.

The development boards ship with the hardware test (hwtest.obj) file already resident. A detailed description of this software bundle is included in Appendix B.

The following table is a specific listing of the files contained in the 167_disk.zip file including the directory to which the file automatically extracts them.

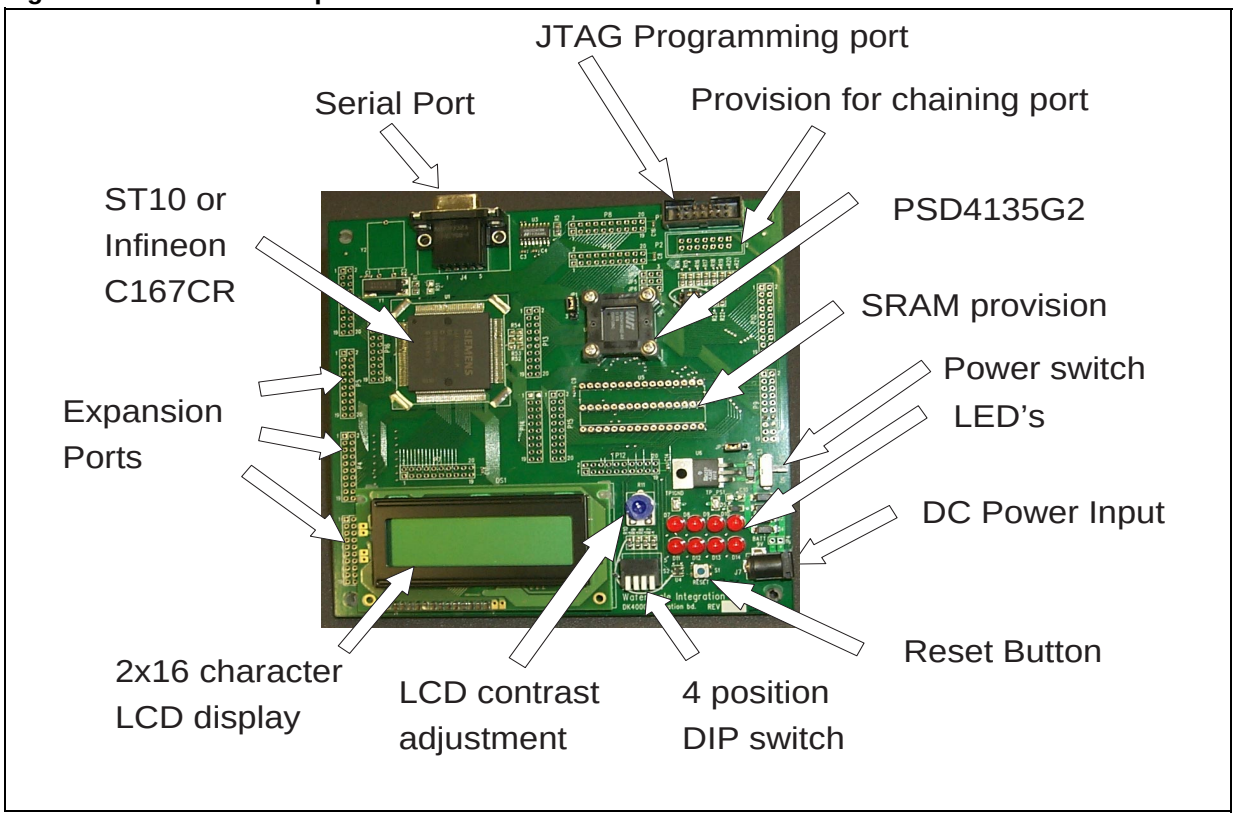
Table 1. Listing of the files and destinations in 167_disk.zip

Directory	Files	Description
root	demo10.h86	alternate firmware to load to demonstrate IAP
	iap.mmf	memory map file for PSDload
	iap.obj	object file to load directly into PSD
	iap.psd	configuration file for psdload
hwtest-167		
	166p_hwt_10s_.zip	contains all PSDsource files
	166c_hwt_10s_.zip	contains all c level code files
	readme.txt	late breaking information
	hwtest.obj	object file to load directly into PSD
demo-167		
	u67eemop10_.zip	contains all PSD source files
	u67democ10_.zip	contains all c level code filed
	readme.txt	late breaking information
	a16xbhe.obj	object file to load directly into PSD
iap_167		
	166p10iap.zip	contains all PSD source files
	166c10iap.zip	contains all c level code files
	readme.txt	late breaking information
	iap.mmf	memory map file for PSDload
	iap.obj	object file to load directly into PSD
demo_iap_167		
	166c10demo.zip	contains all c level code files
	demo10.h86	alternate firmware to load to demonstrate IAP
	readme.txt	late breaking information

Note: 1. Hex file carries the extension *.h86 from the Keil tools.
2. Infineon Dave 2.0 cd was used in these projects
3. Keil compiler version is 4.03 or later. See readme file for particulars.

DETAILED DESCRIPTIONS

Figure 1. DK4000 Development Board



The following features are included in the development board and shown graphically in the above figure.

- Display - A two line by 16 character LCD display is included on the Development Board.
- Power switch
- UART Serial Port(female) - Connected to MCU serial port; used for In-Application Programming (IAP)
- Infineon C167CR or other MCU
- PSD4000 software - The PSD4000 is programmed with C167CR demonstration code. User can program alternative programs via JTAG ISP.
- JTAG programming Port - Used in conjunction with FlashLINK programmer for ISP.
- Reset Button - For resetting the MCU and PSD
- Pads for additional SRAM - The resident PSD4000 contains 8KB SRAM. This site is for additional SRAM.

OTHER BOARD FEATURES

Other features of the DK4000 board are listed below. These elements unpopulated to provide lowest cost to the user.

- Provision for C167CR boot elements are provided with JP5, 6 and 7.
- Provision for chaining JTAG connector is provided in P2 and JP2.
- Provision for C167CR OWE control is provided in JP8
- Provision for an analog Vref input is provided in JP9

- Provision for off expansion is provided by board connectors suitable for 0.025 square posts
- Provision for 9v battery input is provided near power connector (solder pads only).

Step-By-Step Instructions for ISP Programming

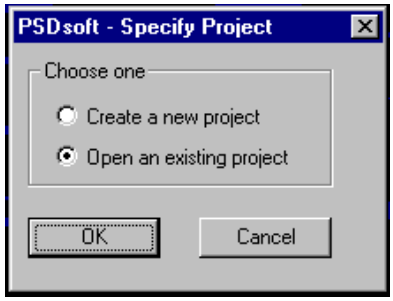
a) Install PSDsoft Express on your PC running Windows 95/98/NT/2000. Check web for latest version.

b) Plug the FlashLINK Programmer into your PC's parallel port and plug in the ribbon cable to the JTAG port on the eval board (for help, see the Appendix C, FlashLINK manual).

c) Plug in power supply and turn on power. An LCD contrast control is provided as R11. The typical setting is near the counterclockwise stop.

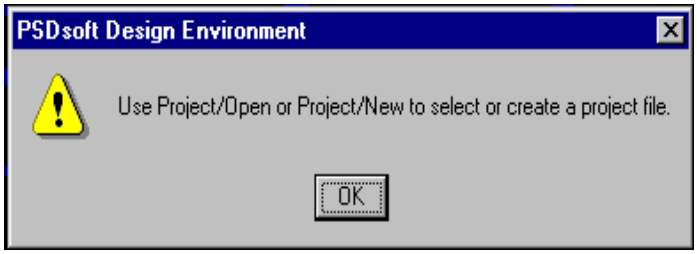
d) Run PSDsoft Express. Here is the initial screen if no project was open.

Figure 2. Opening screen upon PSDsoft Express invocation



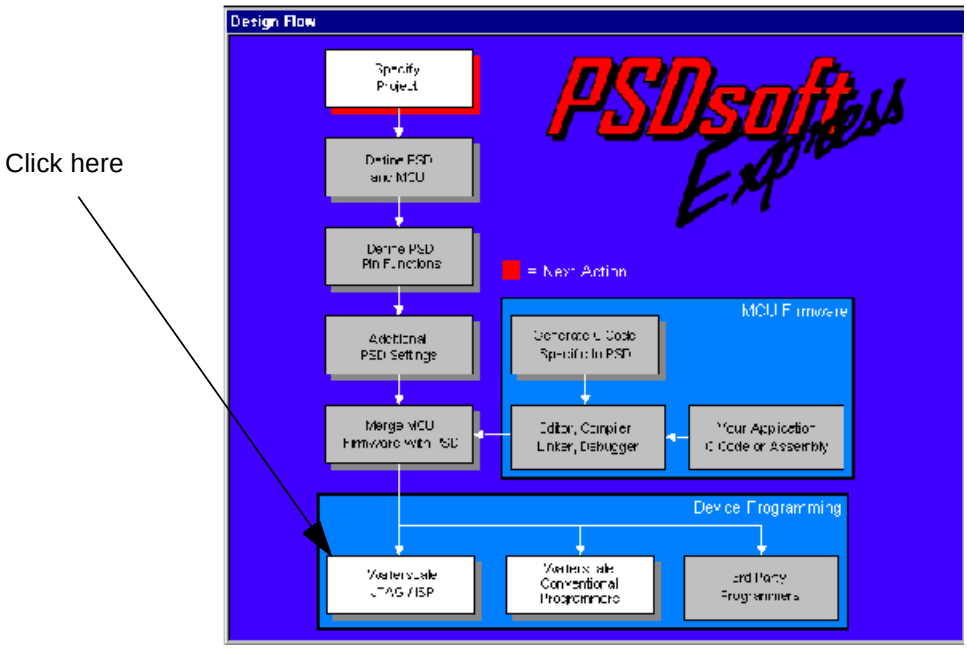
Use cancel at this point since all we need to do is program the PSD with an existing demonstration file (*.obj) and there is no need to create a new project. Later, in the "Using the DK4000 as a development platform", a further tutorial is given on using PSDsoft Express with the Eval Board for development.

Figure 3. Invocation reminder screen



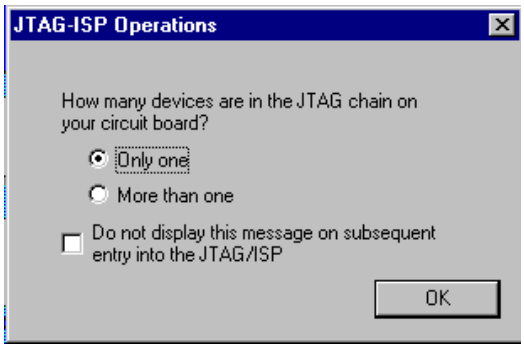
e) In the Design Flow (shown below), click on the ST JTAG/ISP button. Bottom row of boxes left side.

Figure 4. PSDsoft Express flow



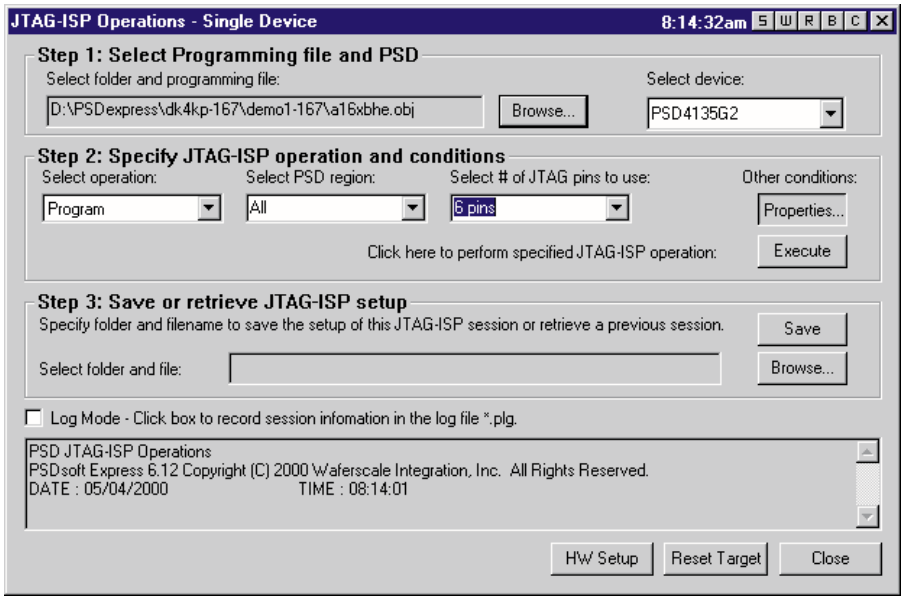
The following screen appears inquiring if it's desired to program a single device or multiple devices in the JTAG chain. Select "Only one" and then click **OK**.

Figure 5. JTAG-ISP Operations dialog



Clicking **OK** brings up the JTAG Operations - Single device dialog shown in the following figure.

Figure 6. PSDsoft Express, JTAG Operations dialog



AI05709

f) In Step 1, browse to find the *.obj file shown in the above figure

g) In “select device” box, choose the PSD4000 device you have installed on the board

h) In step 3, select the operation of “Program”. Click execute.

i) Observe in the lower pane the JTAG activities that occur while programming your device.

j) Watch the display. When the download is completed as indicated in the log window, push the reset button on the Development Board. The displays below will sequence one time and then operation will stop.

Figure 7. Eval Board Displays for IAP.obj

If you cycle power to the board, you will see that the display will resequence, confirming that the program and all configuration information are stored in the PSD's non-volatile Flash memory.

k) For better understanding of the program you may want to examine the following references:

1. System memory map in the "167 Design Overview" section of this document.
2. PSDsoft Express project (a16xbhe.ini)
3. The file source code (included) to see how the executing code was configured

Notice:

An additional code bundle will be posted on the web in the future to cover the IAP functionality. Please go to *www.st.com/psd*, and select "Development Tools" and scroll down to DK4000. Latest software and manual can then be downloaded.

USING DK4000 AS A DEVELOPMENT PLATFORM FOR C167CR USERS

Concept

- The ST DK4000 Development Board provides the following capabilities
- Demonstrate design concepts early, optimizing "time to market"
- Jump start user application with proven framework (hardware and software)
- Substitute for user target system until target prototypes are available
- Gives instant platform for testing ISP and IAP demonstration.
- Allows programming the PSD using included Flashlink cable

Downloading to the Development Board

Executable code can be downloaded to the Development Board two different ways; via the JTAG (ISP) or via the UART (IAP). This manual only describes the ISP capabilities at this time. The IAP capabilities will be supported in the future using PSDload available on the website at *www.st.com/psd*.

JTAG - ISP

The PSD4000 series JTAG interface provides the capability of programming all memory areas within the PSD (PLD, configuration, MAIN and secondary Flash memories). This interface can also be used to program a completely blank component as JTAG is enabled as the default PSD state. See Application Note 54 (AN054) for further description of the JTAG interface on our CD or our website at *www.st.com/psd*.

The LCD will be non operational during JTAG - ISP, since the MCU is not operating. During this interval, the PSD is not connected to the MCU bus. To restrain the MCU during this interval, the JTAG interface contains a signal,!RST, that is connected to the MCU reset pin.

ST provides a FlashLINK programmer to facilitate the JTAG programming operation. The FlashLINK programmer connects the PC parallel port to the Eval Board JTAG header and is driven by PSDsoft Express, the PSD development tool.

167 DESIGN OVERVIEW

The following figure depicts how the memory is allocated in this project for the hwtest.obj. Hwtest.obj uses the segmented mode of the C167CR. The demo1 project uses the non-segmented mode of the C167CR.

The C167CR contains a large addressable memory area that is partitioned into segments of 64k bytes each. Even though many memory segments exist in the C167CR, only segments 0 and 1 are used in this

project.

The configuration of the C167CR is controlled by two registers that are written at system startup (syscon and buscon0). These registers handle the mechanism for different bus width peripherals in the C167CR as well as many other items. See the C167CR user manual for details. Additional areas can be controlled by the definition of pairs of buscon and addr registers for each discrete area. The project definitions of these registers are denoted in the memory map figure below.

The default configuration (syscon and buscon0) is 16 bit multiplexed for the following system resources;

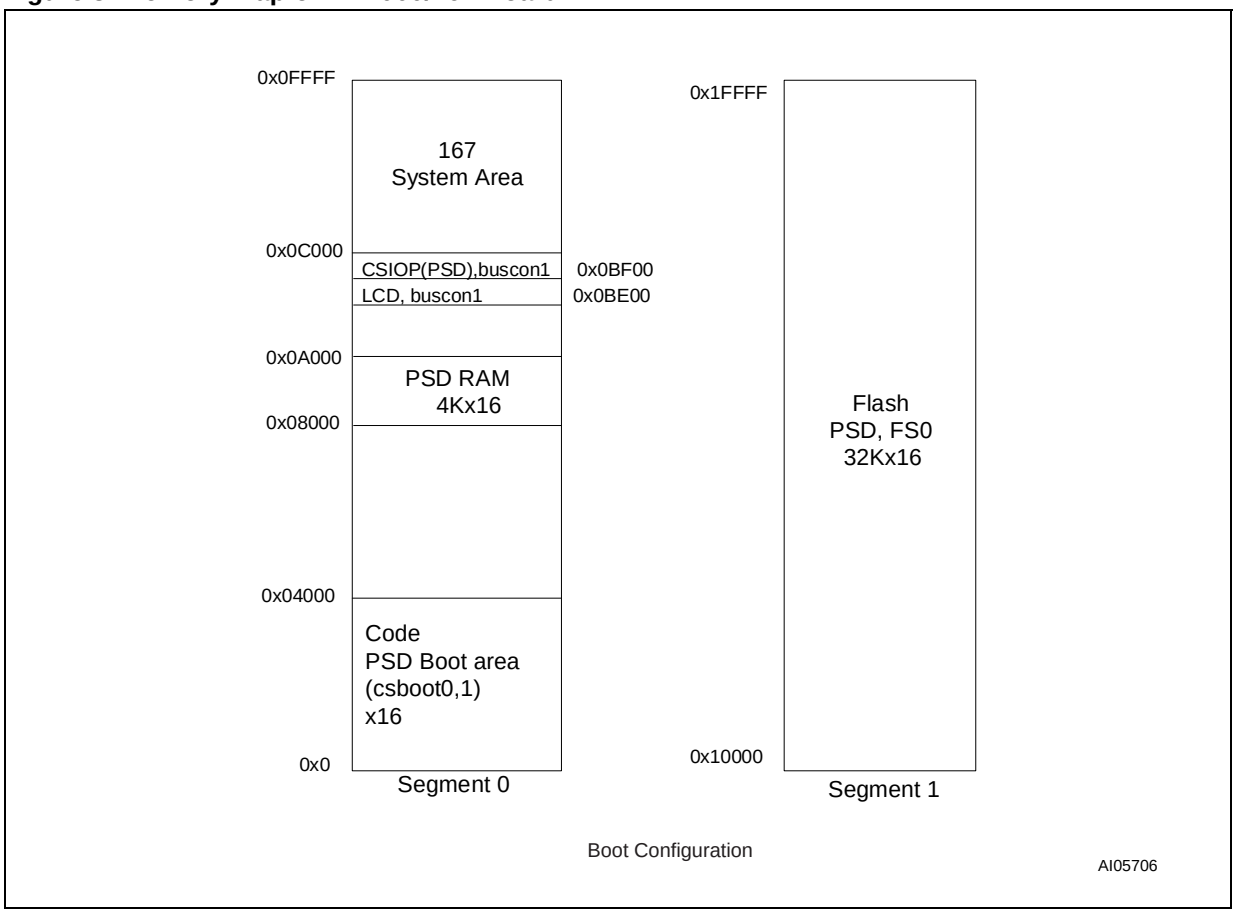
- C167 resources
- PSD code memory (main and secondary Flash memory and boot areas)
- PSD SRAM

Two additional areas are defined as 8 bit multiplexed as shown below for the following system resources.

- LCD
- CSIOP space (PSD registers).

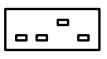
The C167CR XRAM and CAN areas are not used.

Figure 8. Memory map of DK4000/167 Board



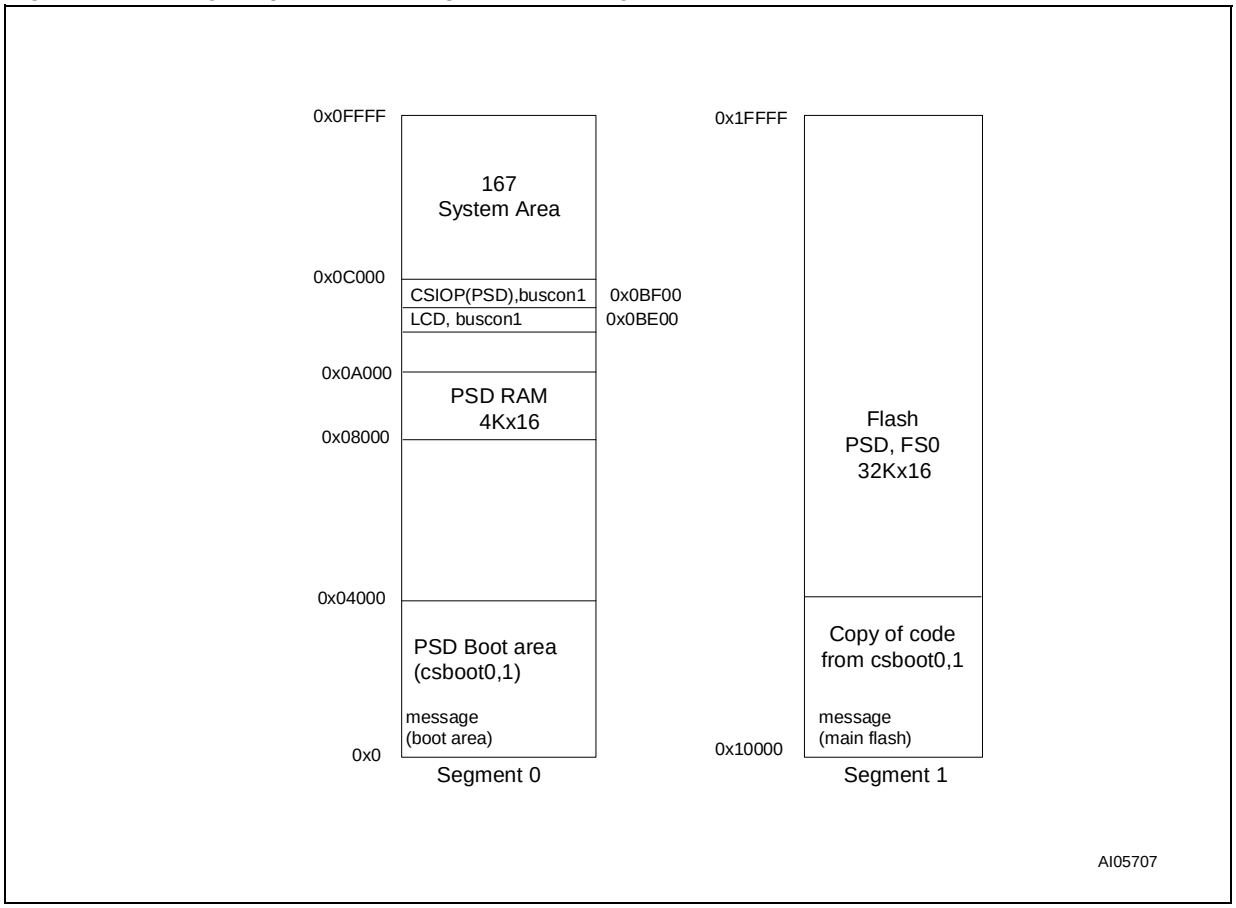
Note: Default x16 multiplex bus used (syscon and buscon0) unless otherwise noted

Memory Swapping in the PSD

For this test (hwtest.obj), the dip switch should be in the following position  . As a component of this test, a copy of the executing code that resides in csboot0/1 is made. The destination of this copy is

the main Flash memory area FS0, as shown in the figure below. After the copy operation, the following map applies.

Figure 9. Memory map after running of hwtest.obj



Notice in the above figure the element denoted as *message* in each of the code areas. This element is displayed as the second LCD screen to show the source of execution. For normal boot, the second LCD screen shows “executing from, BOOT area”. The message exists in a fixed location in the code and is read from this location and copied to the LCD at boot up.

When the code copy is performed, a different message is inserted into the same fixed location based on the destination of the copy (as shown in FS0). When this version of the code is executed, the message is displayed “executing from MAIN FLASH”. This method yields a single unambiguous confirmation of the execution source, which is very convenient for demonstrating memory swapping operations.

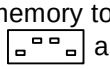
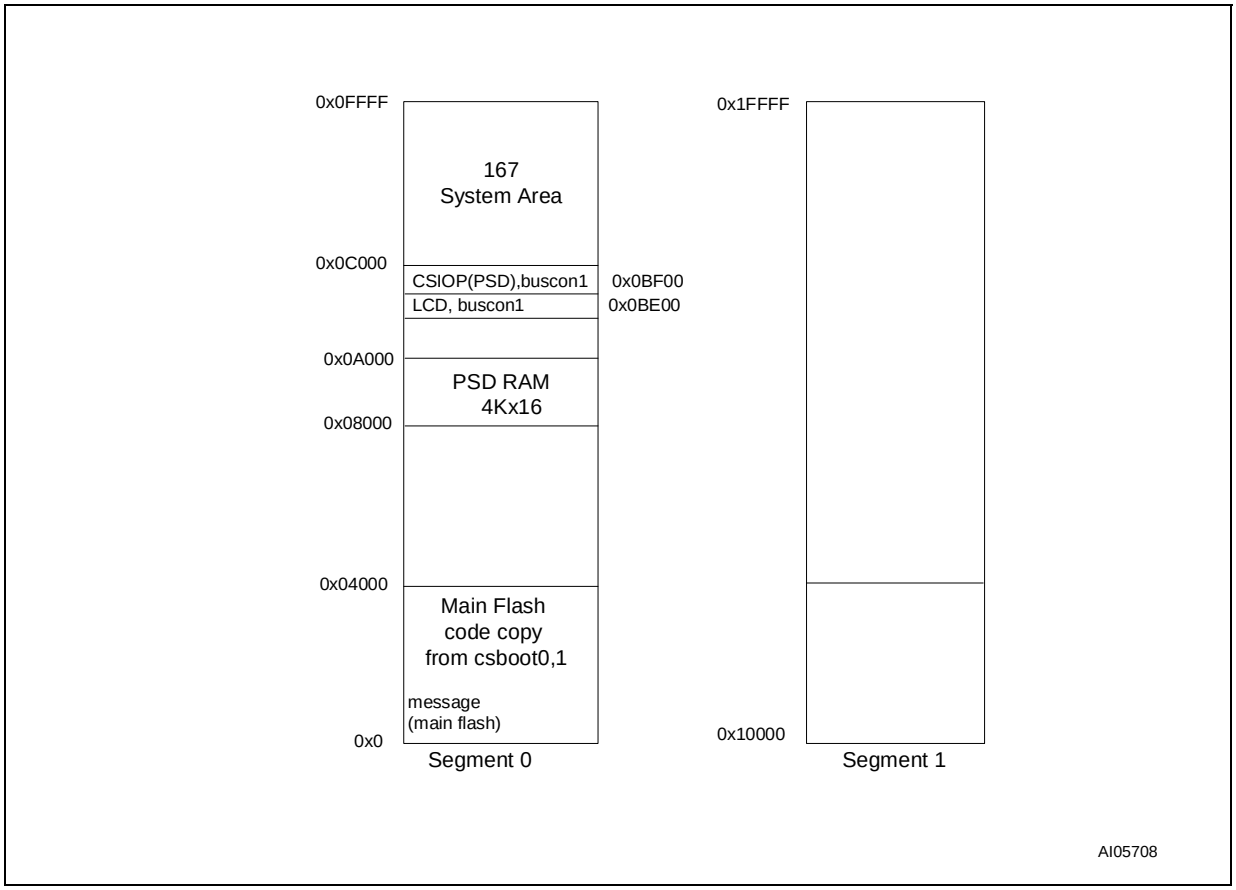
Now let's boot from the other memory to demonstrate the swapping capability of the PSD. Place the dip switch in the following position  and press the reset button. You should see the execution source annunciated to the display "booting from MAIN FLASH". The following memory map applies.

Figure 10. Memory map for alternate memory boot



The memory movement within the MCU memory map is accomplished via the logic contained in the PLD equations in the PSD. Each segment that moves must have dual ranged defined in these equations. The selection is made based on a single logic bit (exe_src_a) that resides in the PSD PAGE register. Following are the equations for the system. These can bee seen in the PSDsoft Express project included with the kit. Note that "#" indicates a logical OR and "&" indicates a logical AND

```
Csboot0 = (0x0 – 0x01FFF) & !exe_src_a
Csboot1 = (0x02000 – 0x03FFF) & !exe_src_a
Fs0 = ( (0x10000 – 0x1FFFF) & !exe_src_a )
        # ( ( 0x0 – 0x03FFF) & exe_src )
```

Note that the logic variable (bit) controlling the actual location of the memory is "exe_src_a". When this bit is zero (0), the memory segments are as shown in figure 9. When exe_src_a is one (1), FS0 appears in the execution location and the csboot areas are not in the map at all. The physical location of this logic bit, exe_src_a, is in the bit6 position of the PAGE register. Actually this bit can be anywhere, the only important element is that it is contained in the PLD equations as shown above and accessible by the MCU. Control of this bit is via a board mounted dip switch.

The power up sequence is as follows:

1. Execute start167.a66

2. Read the dip switch

3. Write the dip switch setting into the PSD PAGE register (some positioning is done prior to the write).

Once the PAGE register write operation has completed, the next instruction is fetched from the new memory location (FS0).

This same sequence of events occurs every time power is applied to the board. Since the PAGE register is always 00h at power up, the software always executes steps a) and b) from the boot area. Then, based on the dip switch selection, the code will either stay in the boot area or jump to the main Flash memory area.

What really happens

There is a subtlety involved in the transfer of execution described above. This subtlety is because the MCU really doesn't know the source of the instruction bytes; boot area or main Flash memory. All the MCU knows is that valid instructions on valid address boundaries are presented on the bus when the MCU needs them. Then the MCU executes the instruction and generates the next address. The key element involved is the generation of the address by the MCU.

To understand this critical transfer of control, let's examine the instruction by instruction transition from one memory to the other. After the reset signal is deasserted, the MCU is executing from the csboot area normally. This continues until the exe_src_a bit is written, moving FS0 into the execution location (0x0-0x3FFF). At this same time, csboot area is, for all practical purposes, gone from the system memory map. At this point, the MCU is generating the next address from the instruction received from the csboot area. However, the next instruction will come from the FS0 area. This next instruction fetch must be appropriate to maintain the program flow. That is, the next instruction must be received by the MCU on an instruction boundary and be appropriate for the program flow. In addition, any issues with the stack and stack pointer must be resolved so program flow can continue (subroutine return addresses, temporary variables, etc.). Pipelining operations can result in execution from the pipeline instead of the new memory, but the pipeline will continue to be filled from the new memory.

The method we've used to ensure correct operation is to place identical code at identical locations in both applications through the point of the swap. After the point of the swap, the code bundles can diverge without problems. While this result is inherently ensured in a code copy scenario like hwtest.obj, it's not so automatic when the applications are different such as exists in a true IAP scenario.

Creating your own IAP code bundle

A few easy steps can ensure that program flow for this critical area is guaranteed to occur properly. These steps involve the absolute location of certain modules within the base application and the new IAP application. Locating these modules is accomplished using linker controls. With this framework, booting from one application to another is EASY.

REFERENCES

IEEE Std 1149.1-1990 IEEE Test Access Port and Boundary Scan Architecture

Flashlink User Manual (included in the Appendix of this document)

AN1153 Application note: JTAG Information

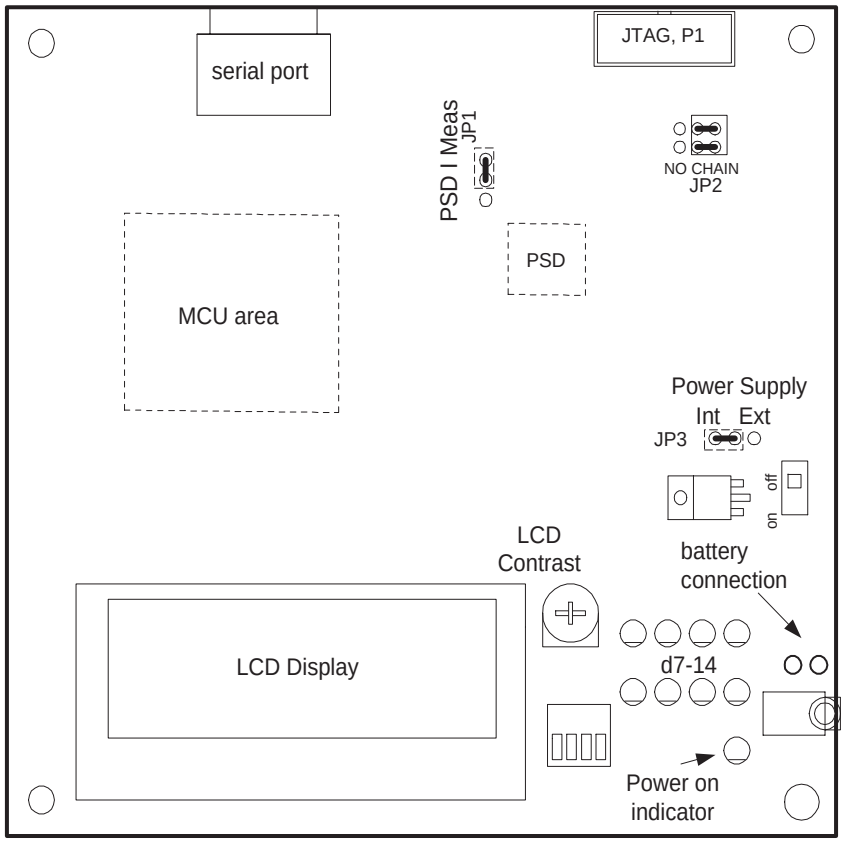
AN1426 Application note: Design Guide, PSDsoft Express and PSD4135G2

APPENDIX

Appendix A - Jumper configuration on DK4000.

Jumper	Description	Default position (shown by dotted line)	Board position
JP1	Measure PSD current	No measure	Upper center
JP2	JTAG chaining	No chain	Upper right
JP3	Internal / external power supply	Internal power supply	Lower right
JP4	9v battery connector	None (no jumper)	Lower right

Figure 11. Assembly Drawing with default jumper positions



Appendix B Software functional description.

Hwtest. This code exercises all components of the development board; the display, PSD memory and chip selects, and the UART channel (single character only on receive and transmit). This confirms functionality and is used as a production test. The following list is a detailed description of the viewable LCD screens.

- Invocation banner, software version
- Display execution source. (boot area or main Flash memory)
- Motherboard LED test
- PSD RAM test

- Code Copy. Executing boot code is copied to main Flash memory block FS0 (BOOT-> FLASH)
- Displays Flash memory ID and does erase of FS0 prior to the copy operation.
- UART test (waiting for host to send “0”, development board reply is a “1”, baud rate is19200 with 8 data bits, no parity and one stop bit)

This software also includes provision for external SRAM test in evaltest.c but this code is not utilized at this time since the site is unpopulated. If this code is used, be aware that the appropriate buscon must be set for 8 bit demuxed bus.

After this code has run one time, a copy of the executing code exists in the Flash memory area (FSx). The system can run from this code copy by placing the dip switch in the appropriate configuration as described in the “Memory Swapping in the PSD” section of this document.

Demo1. This is a simple program that displays the following text on the LCD display.

No Need to fear, EASYflash is here!!!!.

The intent is to show a minimal level of functionality. No UART support is provided.

Appendix C Development Board Schematic and parts list

Figure 12. Main Schematic

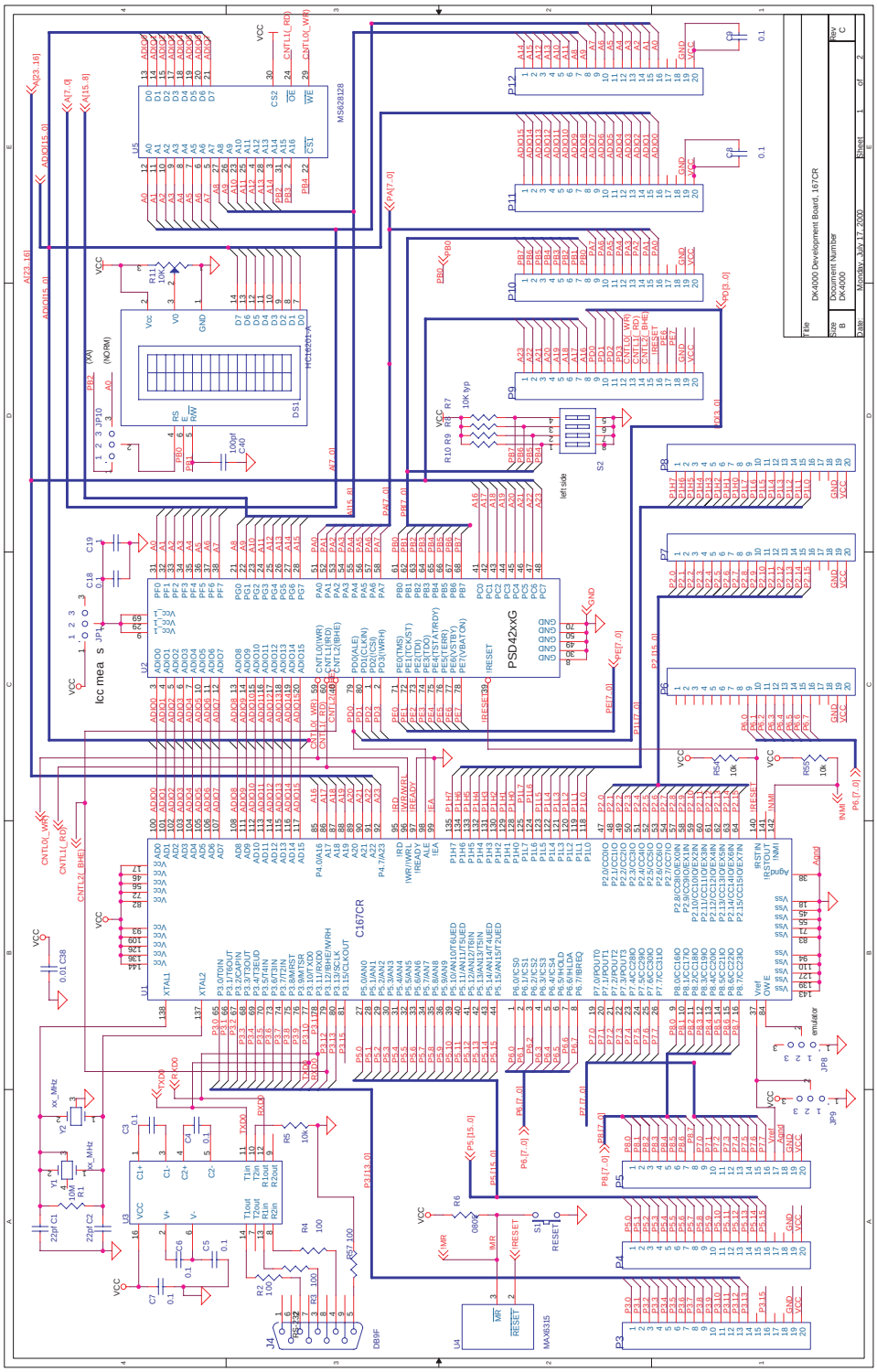


Figure 13. Power Supply Schematic

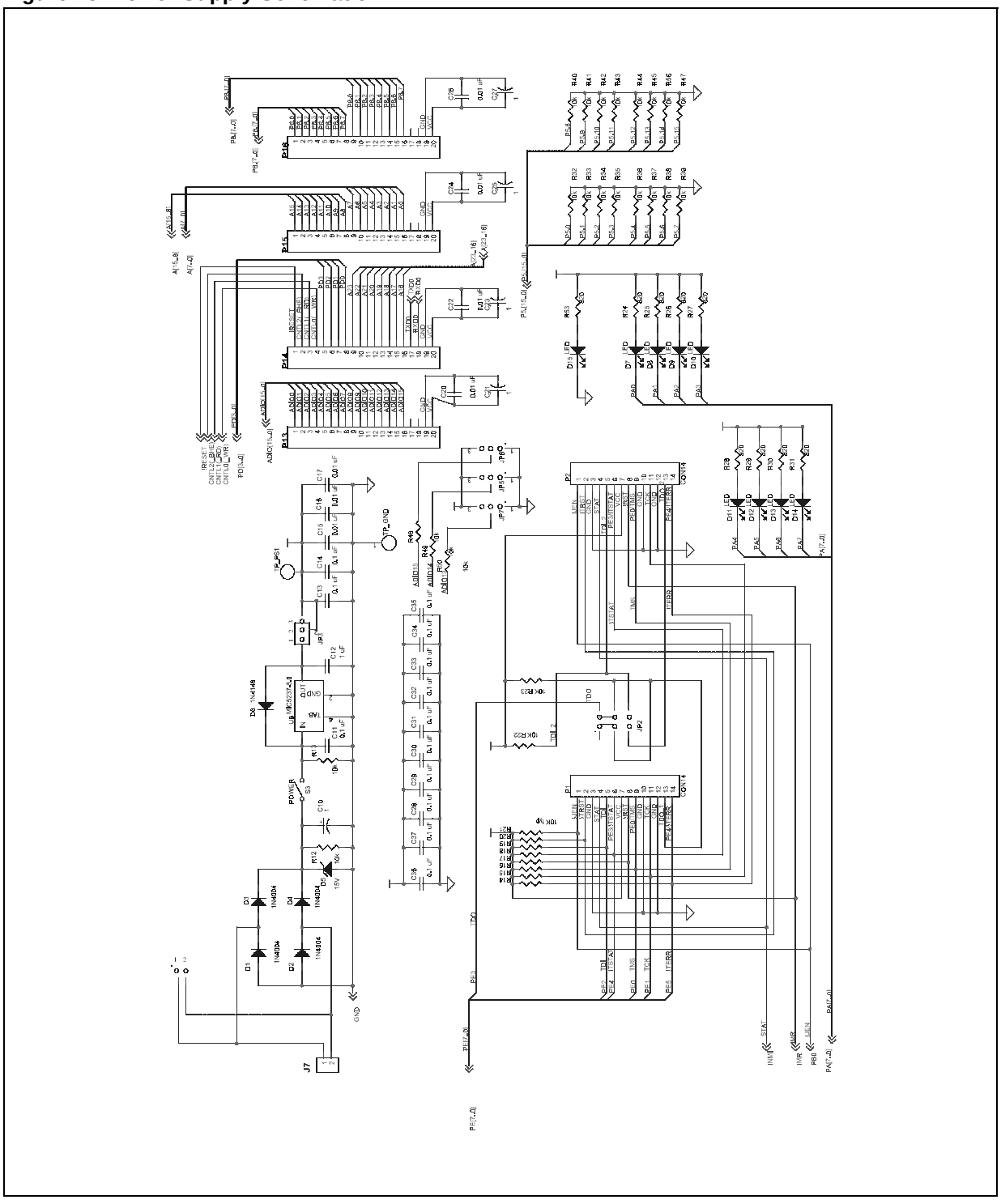


Table 2. DK4000 Parts List

Ref Des	Quantity	Generic Part Number	Description	Vendor	Part Number
	1	pcbevm0002	asian 21insq, us 25 in sq@		
ds1	1	dis101-0001	display	hantronix	hdm16216h-b
y1	1	y101-0002	crystal, 16MHZ		vfsmc-316pf11.0592
u1	1	umcu0002	microcontroller	infineon	C161V
u2	1		PSD42xxG	ST	PSD42xxG
			psd socket	Yamiachi	IC149-080-030-S5
u3	1	u232-0001	232 driver	analog devices	adm202jrn
u4	1	usup0002	max 6315	maxim, 5v	max6315leuk
u6	1	ureg-0001	regulator	micrel	mic5237-5.0bt
d1-4	4	cr101-0001	diode		s1ab
d5	1	vr101-0001	zener diode, 15v	motorola	mmsz5254bt1
d6	1	cr101-0002	signal diode	national	fdLL4148
d7-15	9	led101-0002	led, t5(t1.75)	lumex	SLX-LX5093ID
c1-2	2	cap0805-2209	22 pf caps, cer	murata	grm40c0g22050ad
c10,c21,c23,c25	4	cap1206-1004	cap, 1uf tant	murata	grm42-6y5v105z016ad
c12,c19	2	cap1206-1004	cap, 1uf cer, 1206	AVX	1206zc105mat2a
c3-9,c11, c13-18,c20,c22, c24	18	cap0805-1003	0.1 cap, smt, cer	murata	grm40z5u104z016ad
C40	1	cap0805-1009			
r1	1	res0805-1005	resistor, smt, 10M, 1/8 watt, 0805	samsung	rm10j106ct
r2-4, R57	3	res0805-1000	resistor, smt, 100, 1/8 watt	samsung	rm12j101ct
r5-10, r12-r23, r32-47, R54, R55	35	res0805-1002	resistor, smt, 10k, 1/8 watt, 0805	samsung	rm10f1002ct
r11	1		variable resistor, 10k	digikey	3309P-103-ND
R24-31, R53	9	res0805-8200	resistor, smt. 820, 1/8 watt		rm10f820ct
jp1, jp3	2	con225-1003	3 position header	samtec	tsw-103-23-L-s-LL
j1,j3	2	rec225-1002	shunt (use with jpx above)	samtec	snt-100-bk-g
jp2	1	con225-3003	post 3x3	samtec	tsw-103-23-L-T-LL
j2	1	rec225-3002	triple shunt (use with jp3)	samtec	mnt-103-bk-g
j4	1	con232-0001	rt angle rs232 connector(female, 9 pin)	amp	745988-4
s1	1	sw102-0001	reset switch, momentary	bourns	7914g
s2	1	swdip0004	4 position dip switch,side actuated	cts	195-4mst
s3	0	sw101-0002	on-off switch	digikey	EG1906-ND
t1	1	tr101-0001	class 2 transformer, 500ma,female	digikey	dpd090050-p-5
j7	1	con103-0001	connector for ps, male	digikey	pj-202a

Ref Des	Quantity	Generic Part Number	Description	Vendor	Part Number
	1		7x2 ribbon connector	samtec	
p1	1	con104-2007	jtag connectors	samtec	tst-107-01-L-D-LL
P13-P16	4				
	1	con225-1014	14 pin single in line connector/ spacer (display)	samtec	dw-14-17-T-S-250-LL
	4	std102-0250	standoffs for board	richco	SRS4-5-01
	2	std101-0250	standoffs for display, 0.250	richco	dlcbsat-4-01
tp_ps,tp_gnd	2	tp101-0001	test points	koa	rcw
	2	riv101-0281	rivet	rivet king	c-1 (std tubular rivet)

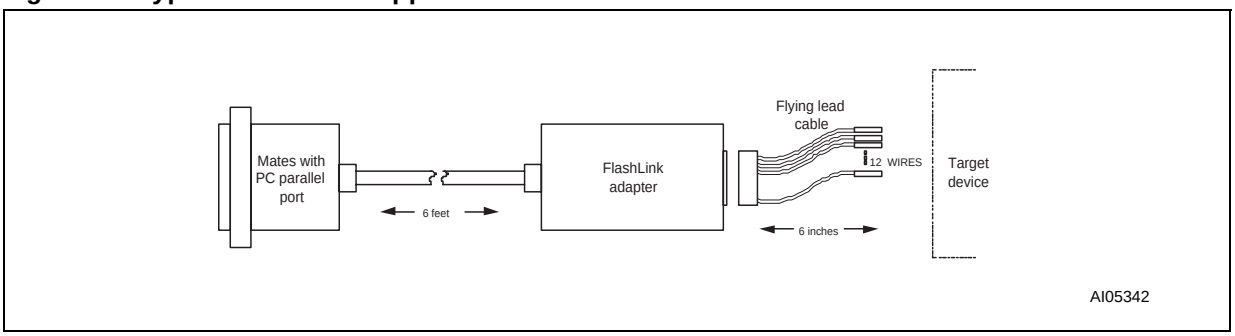
Appendix D: FlashLINK Users Manual

Features.

- Allows PC parallel port to communicate with PSD4000 via PSDsoft Express
- Provides interface medium for JTAG communications
- Supports basic IEEE 1149.1 JTAG signals (TCK, TMS, TDI, TDO)
- Supports additional signals to enhance download speed (!TERR, TSTAT)
- Can be used for programming and/or testing
- Wide power supply range of 2.7 to 5.5V
- Pinout independent with target side flying leads
- Convenient desktop packaging allows varying applications (desk, lab or production)
- Synchronous JTAG interface allows speeds as fast as pc can drive.

Overview. FlashLINK is a hardware interface from a standard PC parallel port to one or more PSD4000 devices located within a target PC board as shown below. This interface cable allows the PSD to be exercised for purposes of programming and/or testing. PSDsoft Express is the source for driving FlashLINK.

Figure 14. Typical FLASHlink application



Operating considerations. Operating power for FlashLINK is derived from the target system in the range of 2.7 to 5.5 V. Compatibility over this voltage range is ensured by the design of FlashLINK. No settings are involved.

On a cautionary note, it is recommended that the target system be powered with a well regulated and stable source of power which is energized at the final value of V_{CC}. It is not recommended that the input voltage be varied using the verneer on a regulated power supply, as this may cause the internal FlashLINK ICs (74VHC240) to misoperate toward the lower end of the supply range.

Each FlashLINK is packaged with a six-inch “flying lead” cable for maximum adaptability (a ribbon cable requires the use a certain connector on the target assembly). This flying lead cable mates to the FlashLink adapter on one end and has loose sockets on the other end to slide onto 0.025 square posts on the target assembly.

Table 3. Pin descriptions for FlashLINK adapter assembly

PIN #	SIGNAL NAME	DESCRIPTION JTAG = IEEE 1149.1 EJTAG = ST EHANCED JTAG	Type	Flashlink is Signal
1	JEN\	Enables JTAG pins on PSD8XXF (optional)	OC,100K	Source
2	TRST\ *	JTAG reset on target (optional per 1149.1)	OC,10K	Source
3	GND	Signal ground		
4	CNTL *	Generic control signal, (optional)	OC,100K	Source
5	TDI	JTAG serial data input		Source
6	TSTAT	EJTAG programming status (optional)		Destination
7	Vcc	VDC Source from target (2.7 - 5.5 VDC)		
8	RST\	Target system reset (recommended)	OC,10K	Source
9	TMS	JTAG mode select		Source
10	GND	Signal ground		
11	TCK	JTAG clock		Source
12	GND	Signal ground		
13	TDO	JTAG serial data output		Destination
14	TERR\	EJTAG programming error (optional)		Destination

Note: 1. Bold signals are required connections
2. all signal grounds are connected inside FlashLink adapter
3. OC = open collector, pulled-up to Vcc inside FlashLink adapter
4. * = Not supported initially by PSDsoft.
5. The target device must supply Vcc to the FlashLink Adapter (2.7 to 5.5 VDC, 15mA max @ 5.5V).

All 14 signals may not be needed for a given application. Here's how they break down:

(6) Core signals that must be connected: TDI, TDO, TMS, TCK, Vcc and GND

(2) Optional signals for enhanced ISP: TSTAT, TERR\

(1) Optional signal to control multiplexing of the JTAG signals: JEN\

(1) Recommended signal to allow FlashLINK to reset target system during and after ISP: RST\

(1) Optional IEEE-1149.1 signal for JTAG chain reset: TRST\

(1) Optional generic control signal from FlashLink to target system: CNTL

(2) Two additional ground lines to help reduce EMI if a ribbon cable is used. These ground lines "sandwich" the TCK signal in the ribbon cable. These lines are not needed for use with the flying lead cable. That is why the flying lead cable has only 12 of 14 wires populated.

FlashLINK pinouts. There is no “standard” JTAG connector. Each manufacturer differs. ST has a specific connector and pinout for the FlashLINK programmer adapter. The connector scheme on the FlashLink adapter can accept a standard 14 pin ribbon connector (2 rows of 7 pins on 0.1" centers, standard keying) or any other user specific connector that can slide onto 0.025" square posts. The pinout for the FlashLINK adapter connector is shown in the following figure.

A standard ribbon cable is good way to quickly connect to the target circuit board. If a ribbon cable is used, then the receiving connector on the target system should be the same connector type with the same pinout as the FlashLINK adapter shown in the following figure Keep in mind that the JTAG signal, TDI, is sourced from the FlashLINK adapter and should be routed on the target circuit card so that it connects to the TDI input pin of the PSD device. Although the name “TDI” infers “Data In” by convention, it is an output from FlashLINK and an input to the PSD device. Also, keep in mind that the JTAG signal, TDO, is an input received by the FlashLINK adapter and is sourced by the PSD device on the TDO output pin.

Figure 15. Pinout for FlashLINK Adapter and Target System

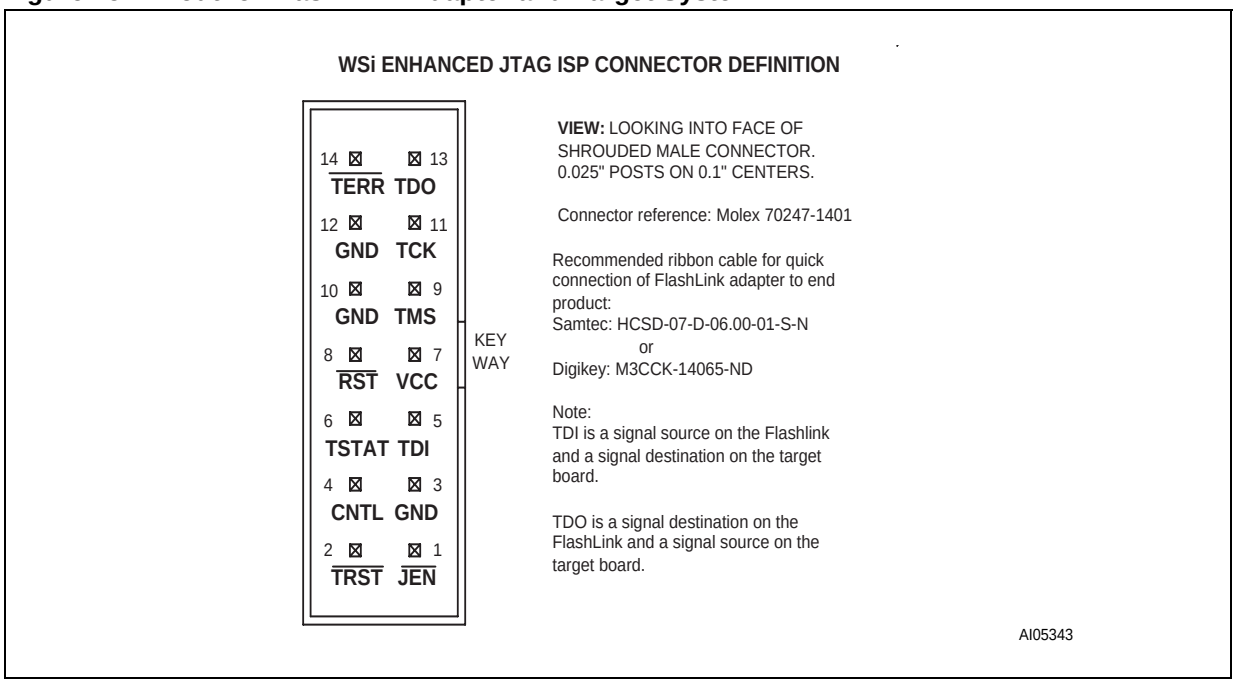


Figure 16. JTAG Chaining Example

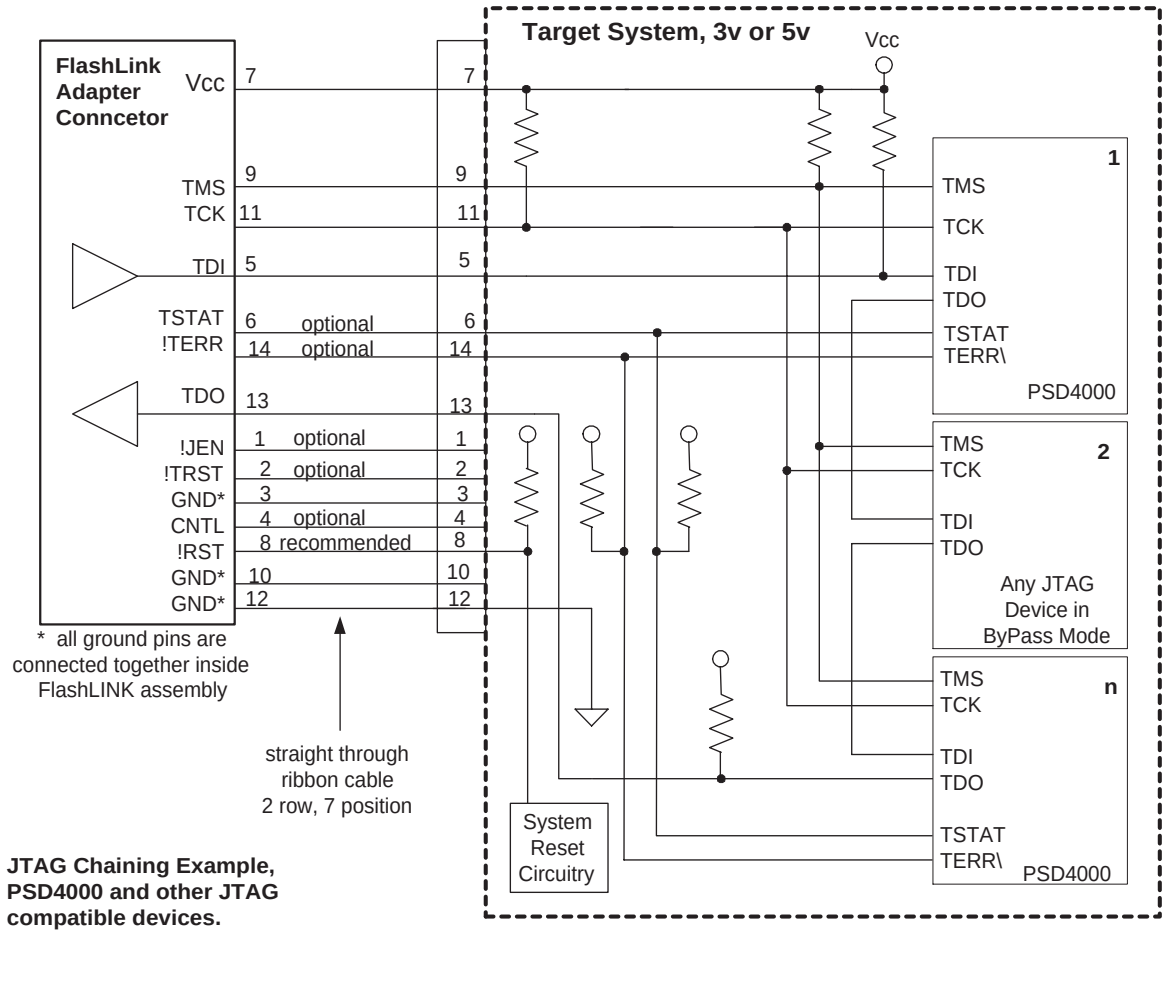


Figure 17. Loop back connector schematic

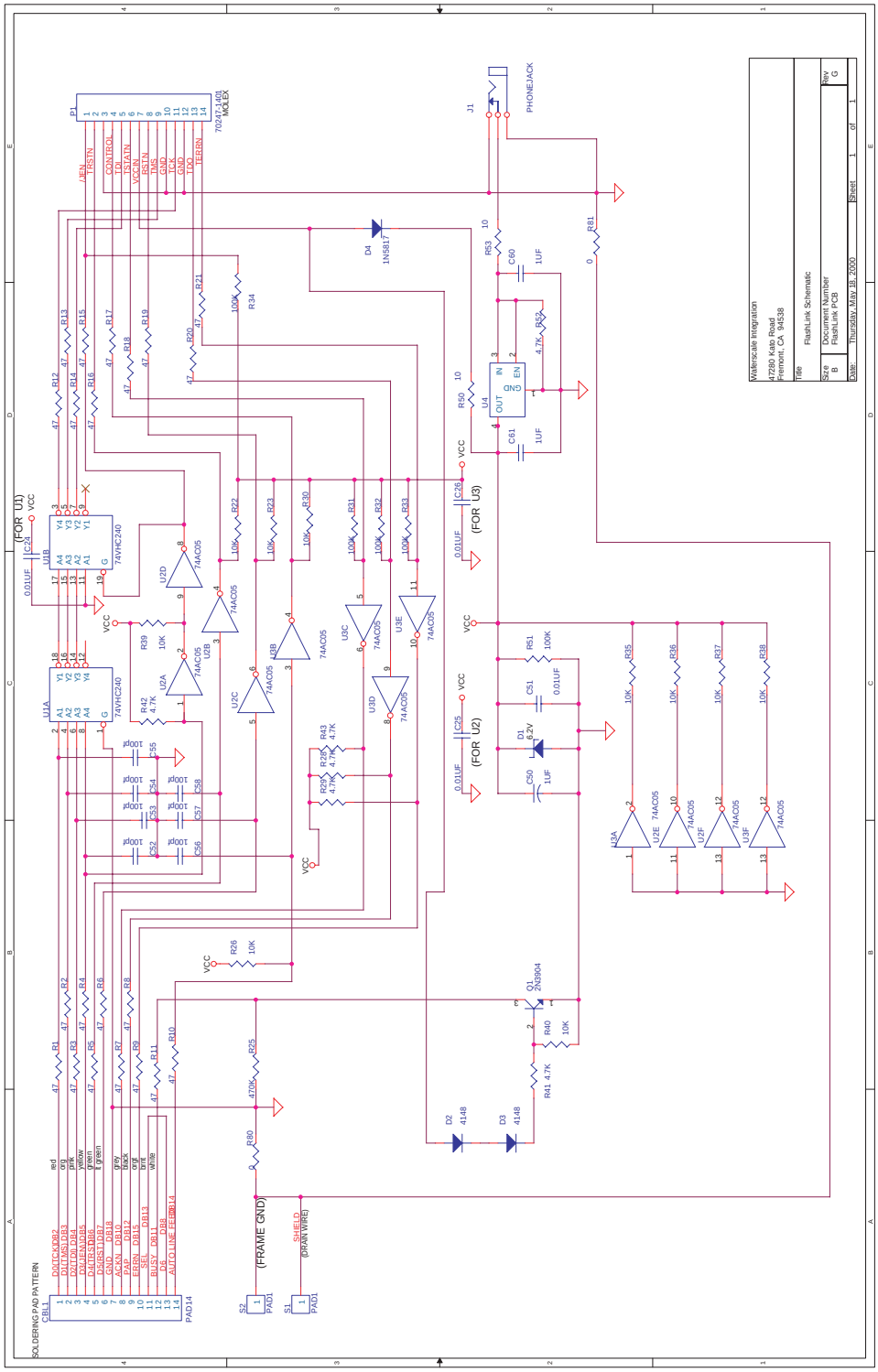
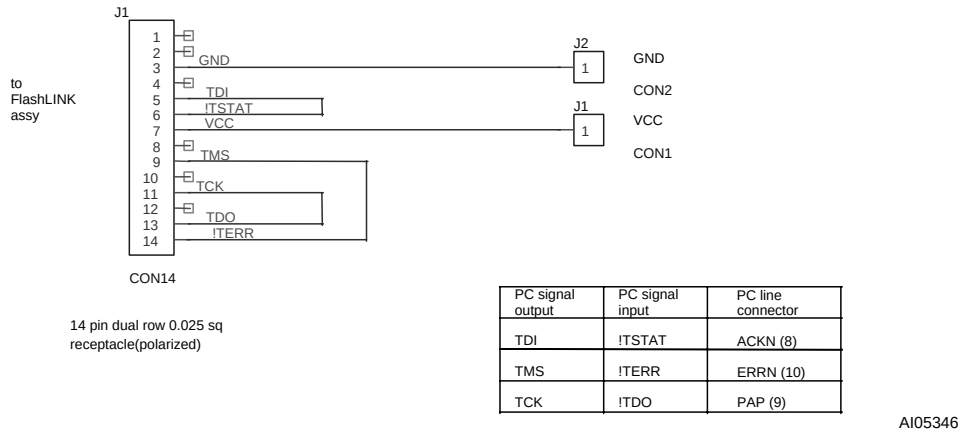


Figure 18. Loop Back Tester, Passive, FlashLINK



Appendix E Results codes and debug tree for 8031_f1.obj

Results codes.

Table 4. Hexadecimal to Binary Conversion

binary		
Results =	abcd	
0	0000	Success Code
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
A	1010	
B	1011	
C	1100	
D	1101	
E	1110	
F	1111	

Table 5. Debug Tree

a	b	c	d		action
x	x	x	1	Page register test	Replace PSD (u1 on EVD) and retest
x	x	1	x	PSD ram error	Replace PSD (u1 on EVD) and retest
x	1	x	x	External Ram error	Replace sram (u3 EVM) and retest
1	x	x	x	Uart error	Repair u4 or surrounding circuitry, EVM (this is under the EVD board)

Note: X = don't care

Appendix F: Board errata

Following is a brief list of issues with correlated on a revision level basis

Rev B. Center row of connections for U5 contain solder mask. This row is intended for a socket to accommodate 0.3" wide SRAM. Also pin 16 is left out. These will be corrected at next board rev.

Table 6. Document Revision History

Date	Rev.	Description of Revision
	1.0	Document written in the WSI format
19-Oct-2001	2.0	Document converted to the ST format

For current information on PSD products, please consult our pages on the world wide web:
www.st.com/psd

If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail addresses:

apps.psd@st.com (for application support)
ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.