Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

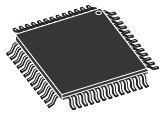
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

UPSD3254A, UPSD3254BV
UPSD3253B, UPSD3253BV

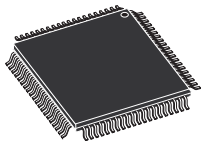
Flash programmable system devices with 8032 MCU and 256 Kbit SRAM

Features

- Fast 8-bit 8032 MCU
  - 40 MHz at 5.0 V, 24 MHz at 3.3 V
  - Core, 12-clocks per instruction
- Dual Flash memories with memory management
  - Place either memory into 8032 program address space or data address space
  - Read-while-write operation for in-application programming and EEPROM emulation
  - Single voltage program and erase
  - 100,000 minimum erase cycles, 15-year retention
- Clock, reset, and supply management
  - Normal, idle, and power down modes
  - Power-on and low voltage reset supervisor
  - Programmable watchdog timer
- Programmable logic, general-purpose
  - 16 macrocells
  - Implements state machines, glue-logic, etc.
- Timers and interrupts
  - Three 8032 standard 16-bit timers
  - 10 Interrupt sources with two external interrupt pins



LQFP52 (T)
52-lead, thin,
quad flat package



LQFP80 (U)
80-lead, thin, quad
flat package

- A/D converter
  - Four channels, 8-bit resolution, 10 µs
- Communication interfaces
  - USB v1.1, low-speed 1.5 Mbps, 3 endpoints
  - I^2C master/slave bus controller
  - Two UARTs with independent baud rate
  - Six I/O ports with up to 46 I/O pins
  - 8032 address/data bus available on TQFP80 package
  - 5 PWM outputs, 8-bit resolution
- JTAG in-system programming
  - Program the entire device in as little as 10 seconds
- Single supply voltage
  - 4.5 to 5.5 V
  - 3.0 to 3.6 V
- ECOPACK® packages

Table 1. Device summary

Order code	Max. clock (MHz)	1st Flash	2nd Flash	SRAM	GPIO	USB	8032 bus	V_{CC} (V)	Pkg.	Temp.
UPSD3253B-40T6	40	128 KB	32 KB	32 KB	37	No	No	4.5-5.5	TQFP52	–40°C to 85°C
UPSD3253BV-24T6	24	128 KB	32 KB	32 KB	37	No	No	3.0-3.6	TQFP52	–40°C to 85°C
UPSD3254BV-24U6	24	256 KB	32 KB	32 KB	46	No	Yes	3.0-3.6	TQFP80	–40°C to 85°C
UPSD3254A-40T6	40	256 KB	32 KB	32 KB	37	Yes	No	4.5-5.5	TQFP52	–40°C to 85°C
UPSD3254A-40U6	40	256 KB	32 KB	32 KB	46	Yes	Yes	4.5-5.5	TQFP80	–40°C to 85°C

Contents

1 UPSD325xx description

The UPSD325xx Series combines a fast 8051-based microcontroller with a flexible memory structure, programmable logic, and a rich peripheral mix including USB, to form an ideal embedded controller. At its core is an industry-standard 8032 MCU operating up to 40MHz.

A JTAG serial interface is used for In-System Programming (ISP) in as little as 10 seconds, perfect for manufacturing and lab development.

The USB 1.1 low-speed interface has one Control endpoint and two Interrupt endpoints suitable for HID class drivers.

The 8032 core is coupled to Programmable System Device (PSD) architecture to optimize the 8032 memory structure, offering two independent banks of Flash memory that can be placed at virtually any address within 8032 program or data address space, and easily paged beyond 64 Kbytes using on-chip programmable decode logic.

Dual Flash memory banks provide a robust solution for remote product updates in the field through In-Application Programming (IAP). Dual Flash banks also support EEPROM emulation, eliminating the need for external EEPROM chips.

General purpose programmable logic (PLD) is included to build an endless variety of glue-logic, saving external logic devices. The PLD is configured using the software development tool, PSDsoft Express, available from the web at **www.st.com/psm**, at no charge.

The UPSD325xx also includes supervisor functions such as a programmable watchdog timer and low-voltage reset.

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com.

ECOPACK® is an ST trademark.

Figure 1. UPSD325xx block diagram

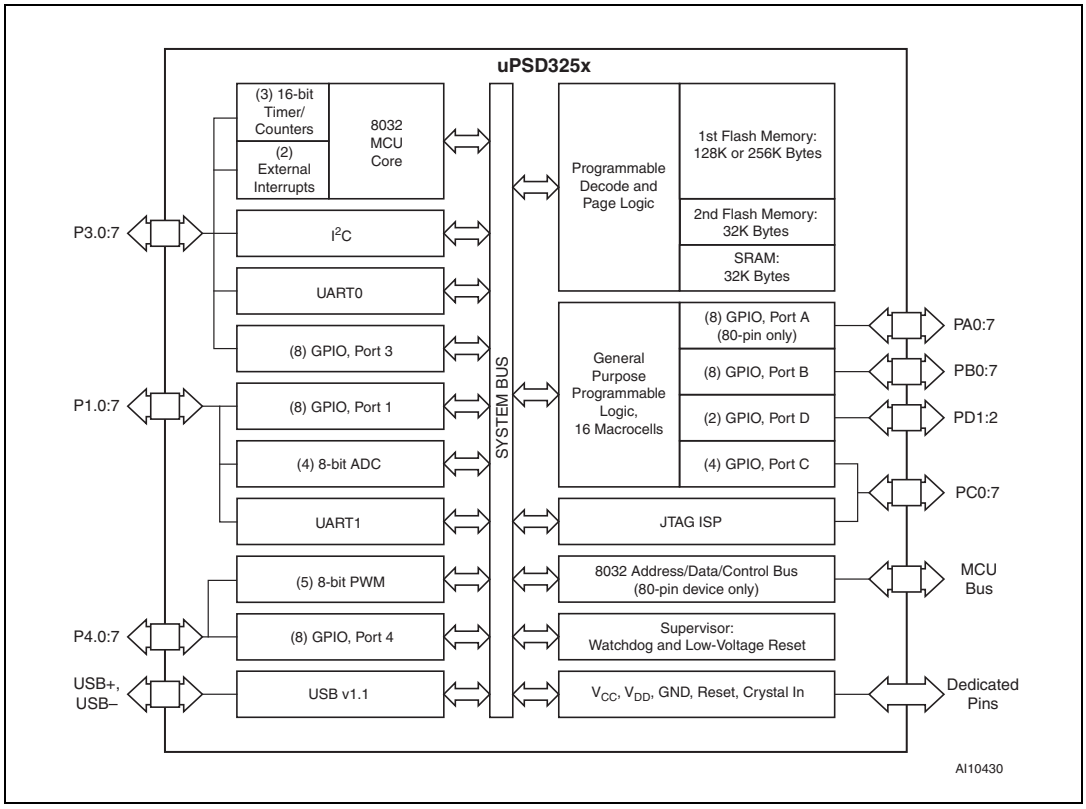
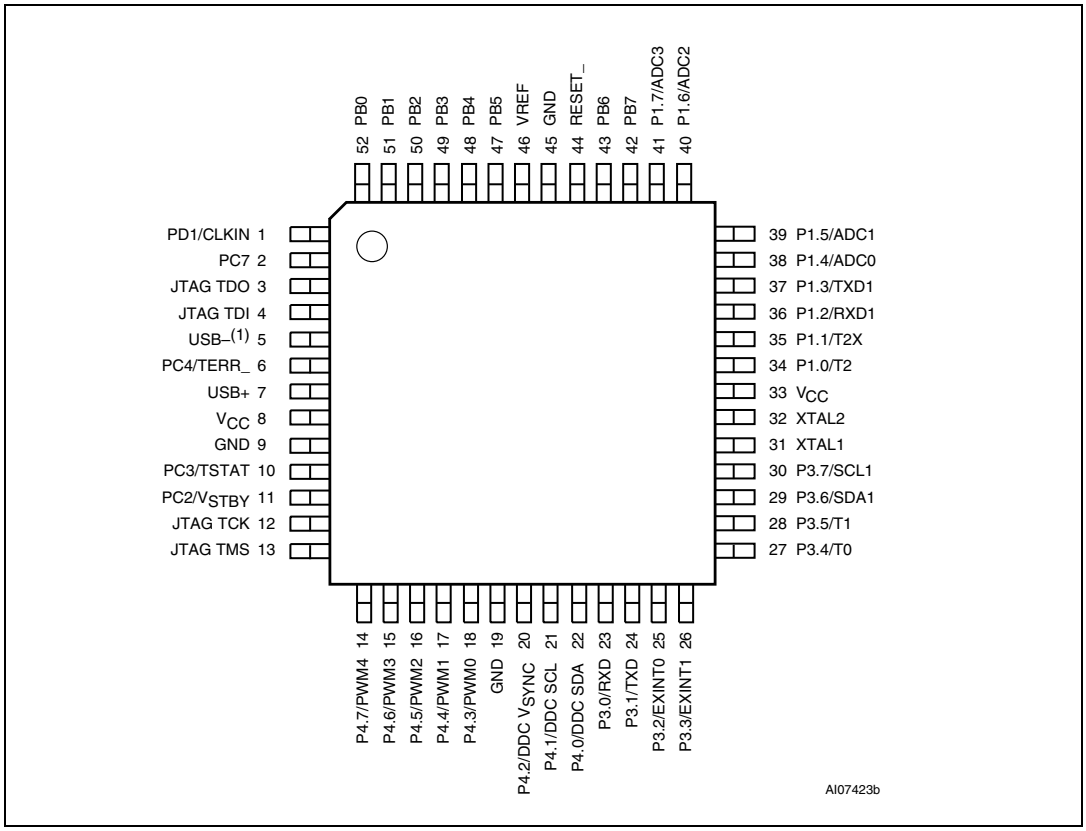
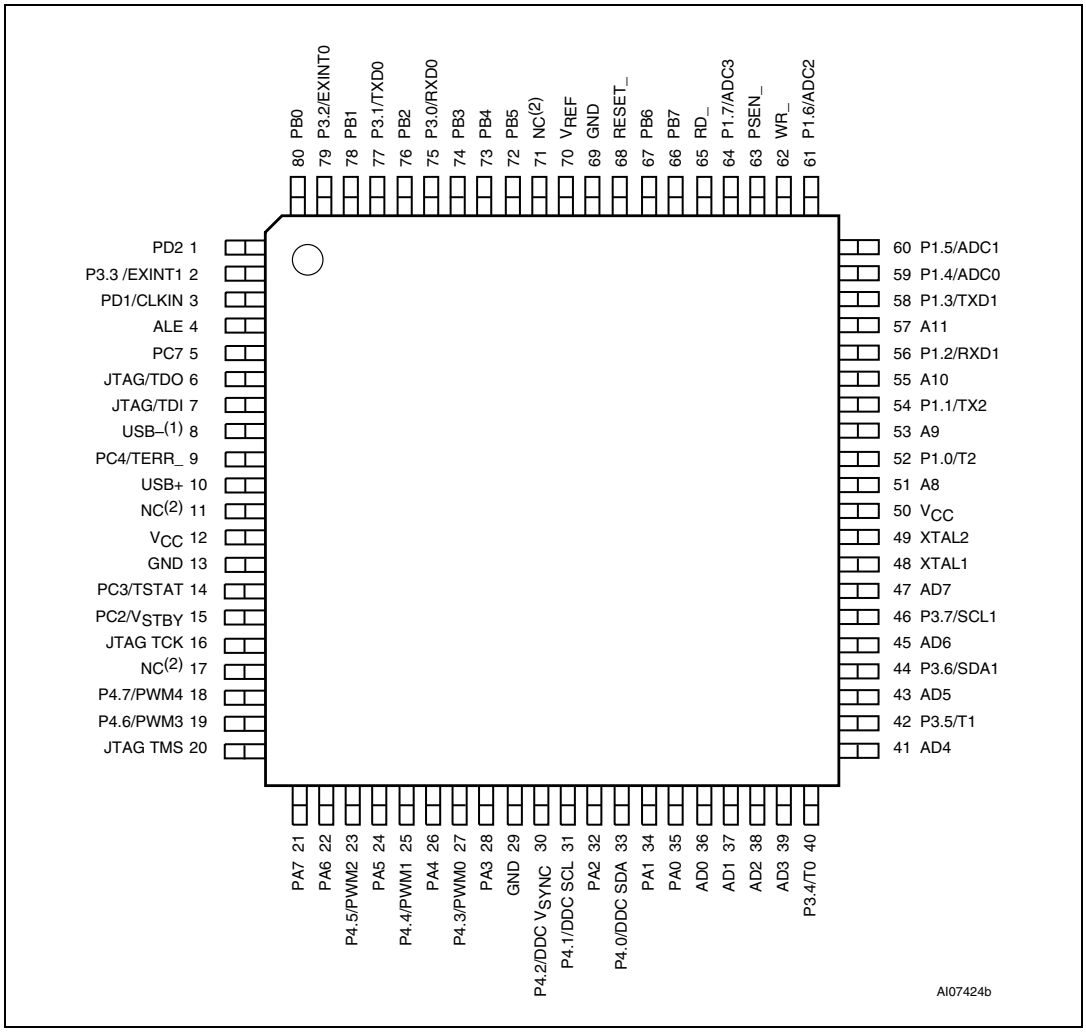


Figure 2. TQFP52 connections



1. Pull-up resistor required on pin 5 (2 kΩ for 3 V devices, 7.5 kΩ for 5 V devices) for all 52-pin devices, with or without USB function.

Figure 3. TQFP80 connections



1. Pull-up resistor required on pin 8 (2 kΩ for 3 V devices, 7.5 kΩ for 5 V devices) for all 82-pin devices, with or without USB function.
2. NC = Not Connected

Table 2. 80-pin package pin description

Port pin	Signal name	Pin no.	In/ out	Function	
				Basic	Alternate
	AD0	36	I/O	External Bus: Multiplexed Address/Data bus A1/D1	
	AD1	37	I/O	Multiplexed Address/Data bus A0/D0	
	AD2	38	I/O	Multiplexed Address/Data bus A2/D2	
	AD3	39	I/O	Multiplexed Address/Data bus A3/D3	
	AD4	41	I/O	Multiplexed Address/Data bus A4/D4	
	AD5	43	I/O	Multiplexed Address/Data bus A5/D5	
	AD6	45	I/O	Multiplexed Address/Data bus A6/D6	

Table 2. 80-pin package pin description (continued)

Port pin	Signal name	Pin no.	In/ out	Function	
				Basic	Alternate
	AD7	47	I/O	Multiplexed Address/Data bus A7/D7	
P1.0	T2	52	I/O	General I/O port pin	Timer 2 Count input
P1.1	TX2	54	I/O	General I/O port pin	Timer 2 Trigger input
P1.2	RxD1	56	I/O	General I/O port pin	2nd UART Receive
P1.3	TxD1	58	I/O	General I/O port pin	2nd UART Transmit
P1.4	ADC0	59	I/O	General I/O port pin	ADC Channel 0 input
P1.5	ADC1	60	I/O	General I/O port pin	ADC Channel 1 input
P1.6	ADC2	61	I/O	General I/O port pin	ADC Channel 2 input
P1.7	ADC3	64	I/O	General I/O port pin	ADC Channel 3 input
	A8	51	O	External Bus, Address A8	
	A9	53	O	External Bus, Address A9	
	A10	55	O	External Bus, Address A10	
	A11	57	O	External Bus, Address A11	
P3.0	RxD0	75	I/O	General I/O port pin	UART Receive
P3.1	TxD0	77	I/O	General I/O port pin	UART Transmit
P3.2	EXINT0	79	I/O	General I/O port pin	Interrupt 0 input / Timer 0 gate control
P3.3	EXINT1	2	I/O	General I/O port pin	Interrupt 1 input / Timer 1 gate control
P3.4	T0	40	I/O	General I/O port pin	Counter 0 input
P3.5	T1	42	I/O	General I/O port pin	Counter 1 input
P3.6	SDA1	44	I/O	General I/O port pin	I^2C Bus serial data I/O
P3.7	SCL1	46	I/O	General I/O port pin	I^2C Bus clock I/O
P4.0	DDC SDA	33	I/O	General I/O port pin	
P4.1	DDC SCL	31	I/O	General I/O port pin	
P4.2	DDC V_{SYNC}	30	I/O	General I/O port pin	
P4.3	PWM0	27	I/O	General I/O port pin	8-bit Pulse Width Modulation output 0
P4.4	PWM1	25	I/O	General I/O port pin	8-bit Pulse Width Modulation output 1
P4.5	PWM2	23	I/O	General I/O port pin	8-bit Pulse Width Modulation output 2
P4.6	PWM3	19	I/O	General I/O port pin	8-bit Pulse Width Modulation output 3

Table 2. 80-pin package pin description (continued)

Port pin	Signal name	Pin no.	In/ out	Function	
				Basic	Alternate
P4.7	PWM4	18	I/O	General I/O port pin	Programmable 8-bit Pulse Width modulation output 4
	USB–	8	I/O	Pull-up resistor required (2 kΩ for 3 V devices, 7.5 kΩ for 5 V devices)	
	V_{REF}	70	O	Reference Voltage input for ADC	
	RD_	65	O	READ signal, external bus	
	WR_	62	O	WRITE signal, external bus	
	PSEN_	63	O	$\overline{PSEN}$ signal, external bus	
	ALE	4	O	Address Latch signal, external bus	
	RESET_	68	I	Active low $\overline{RESET}$ input	
	XTAL1	48	I	Oscillator input pin for system clock	
	XTAL2	49	O	Oscillator output pin for system clock	
PA0		35	I/O	General I/O port pin	PLD macrocell outputs PLD inputs Latched address out (A0-A7) Peripheral I/O mode
PA1		34	I/O	General I/O port pin	
PA2		32	I/O	General I/O port pin	
PA3		28	I/O	General I/O port pin	
PA4		26	I/O	General I/O port pin	
PA5		24	I/O	General I/O port pin	
PA6		22	I/O	General I/O port pin	
PA7		21	I/O	General I/O port pin	
PB0		80	I/O	General I/O port pin	PLD macrocell outputs PLD inputs Latched address out (A0-A7)
PB1		78	I/O	General I/O port pin	
PB2		76	I/O	General I/O port pin	
PB3		74	I/O	General I/O port pin	
PB4		73	I/O	General I/O port pin	
PB5		72	I/O	General I/O port pin	
PB6		67	I/O	General I/O port pin	
PB7		66	I/O	General I/O port pin	

Table 2. 80-pin package pin description (continued)

Port pin	Signal name	Pin no.	In/ out	Function	
				Basic	Alternate
	JTAG TMS	20	I	JTAG pin	PLD macrocell outputs PLD inputs JTAG pins are dedicated pins
	JTAG TCK	16	I	JTAG pin	
PC3	TSTAT	14	I/O	General I/O port pin	
PC4	TERR_	9	I/O	General I/O port pin	
	JTAG TDI	7	I	JTAG pin	
	JTAG TDO	6	O	JTAG pin	
PC7		5	I/O	General I/O port pin	
PD1	CLKIN	3	I/O	General I/O port pin	PLD I/O Clock input to PLD and APD
PD2		1	I/O	General I/O port pin	PLD I/O Chip select to PSD module
Vcc		12			
Vcc		50			
GND		13			
GND		29			
GND		69			
	USB+	10			
NC		11			
NC		17			
NC		71			

1.1 52-pin package I/O port

The 52-pin package members of the UPSD325xx devices have the same port pins as those of the 80-pin package except:

- Port 0 (P0.0-P0.7, external address/data bus AD0-AD7)
- Port 2 (P2.0-P2.3, external address bus A8-A11)
- Port A (PA0-PA7)
- Port D (PD2)
- Bus control signal (RD,WR,PSEN,ALE)
- Pin 5 requires a pull-up resistor (2 kΩ for 3 V devices, 7.5 kΩ for 5 V devices) for all devices, with or without USB function.

2 Architecture overview

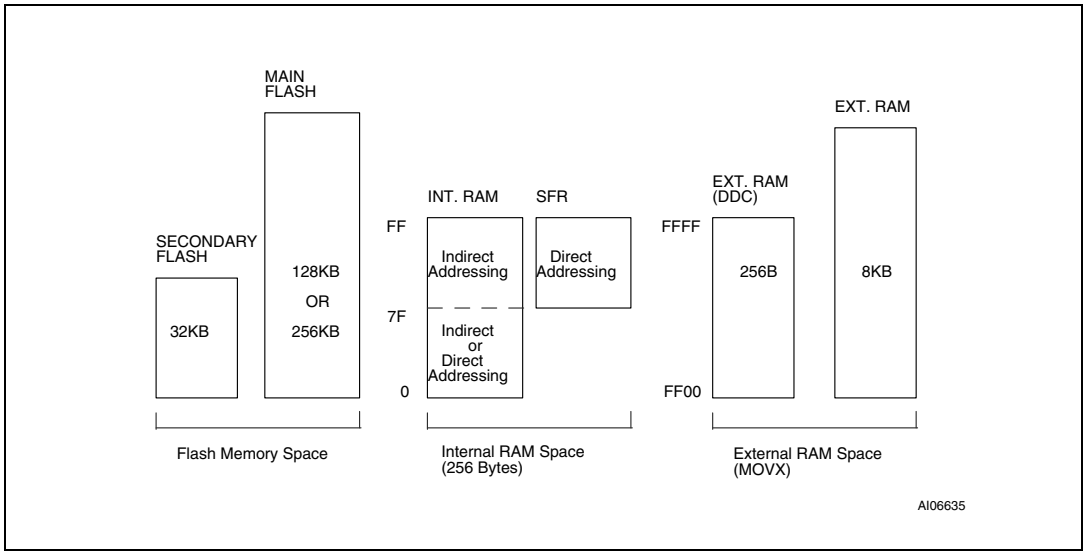
2.1 Memory organization

The UPSD325xx devices' standard 8032 Core has separate 64-Kbyte address spaces for Program memory and Data Memory. Program memory is where the 8032 executes instructions from. Data memory is used to hold data variables. Flash memory can be mapped in either program or data space. The Flash memory consists of two Flash memory blocks: the main Flash memory (1 or 2 Mbit) and the Secondary Flash memory (256 Kbit). Except during flash memory programming or update, Flash memory can only be read, not written to. A Page Register is used to access memory beyond the 64-Kbyte address space. Refer to the PSD module for details on mapping of the Flash memory.

The 8032 core has two types of data memory (internal and external) that can be read and written. The internal SRAM consists of 256 bytes, and includes the stack area.

The SFR (Special Function Registers) occupies the upper 128 bytes of the internal SRAM, the registers can be accessed by Direct addressing only. There are two separate blocks of external SRAM inside the UPSD325X devices: one 256-byte block is assigned for DDC data storage. Another 32 Kbytes resides in the PSD module that can be mapped to any address space defined by the user.

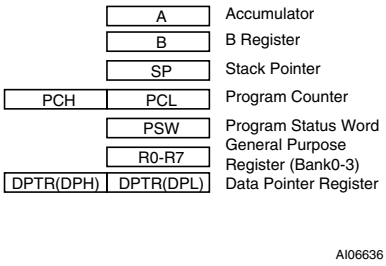
Figure 4. Memory map and address space



2.2 Registers

The 8032 has several registers; these are the Program Counter (PC), Accumulator (A), B Register (B), the Stack Pointer (SP), the Program Status Word (PSW), General purpose registers (R0 to R7), and DPTR (Data Pointer register).

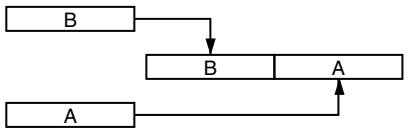
Figure 5. 8032 MCU registers



2.2.1 Accumulator

The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional tests. The Accumulator can be used as a 16-bit register with B Register as shown below.

Figure 6. Configuration of BA 16-bit registers



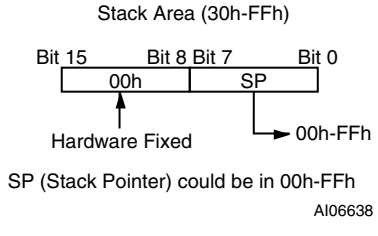
2.2.2 B register

The B Register is the 8-bit general purpose register, used for an arithmetic operation such as multiply, division with Accumulator.

2.2.3 Stack pointer

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07h after reset. This causes the stack to begin at location 08h.

Figure 7. Stack pointer



2.2.4 Program counter

The Program Counter is a 16-bit wide which consists of two 8-bit registers, PCH and PCL. This counter indicates the address of the next instruction to be executed. In $\overline{\text{RESET}}$ state, the program counter has reset routine address (PCH:00h, PCL:00h).

2.2.5 Program status word

The Program Status Word (PSW) contains several bits that reflect the current state of the CPU and select Internal RAM (00h to 1Fh: Bank0 to Bank3). The PSW is described in *Figure 8*. It contains the Carry flag, the Auxiliary Carry flag, the Half Carry (for BCD operation), the General Purpose flag, the Register Bank Select flags, the Overflow flag, and Parity flag.

[Carry flag, CY]. This flag stores any carry or not borrow from the ALU of CPU after an arithmetic operation and is also changed by the Shift Instruction or Rotate Instruction.

[Auxiliary Carry flag, AC]. After operation, this flag is set when there is a carry from Bit 3 of ALU or there is no borrow from Bit 4 of ALU.

[Register Bank Select flags, RS0, RS1]. These flags select one of four banks (00~07H:bank0, 08~0Fh:bank1, 10~17h:bank2, 17~1Fh:bank3) in Internal RAM.

[Overflow flag, OV]. This flag is set to '1' when an overflow occurs as the result of an arithmetic operation involving signs. An overflow occurs when the result of an addition or subtraction exceeds +127 (7Fh) or -128 (80h). The CLRV instruction clears the overflow flag. There is no set instruction. When the BIT instruction is executed, Bit 6 of memory is copied to this flag.

[Parity flag, P]. This flag reflects the number of Accumulator's 1. If the number of Accumulator's 1 is odd, P=0; otherwise, P=1. The sum when adding Accumulator's 1 to P is always even.

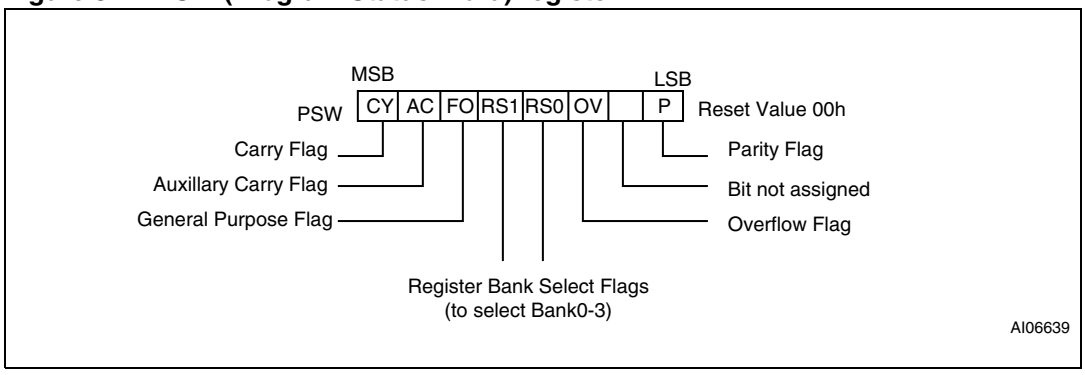
2.2.6 Registers R0~R7

General purpose 8-bit registers that are locked in the lower portion of internal data area.

2.2.7 Data pointer register

Data Pointer Register is 16-bit wide which consists of two-8bit registers, DPH and DPL. This register is used as a data pointer for the data transmission with external data memory in the PSD module.

Figure 8. PSW (Program Status Word) register



2.3 Program memory

The program memory consists of two Flash memories: the main Flash memory (1 or 2 Mbit) and the Secondary Flash memory (256 Kbit). The Flash memory can be mapped to any address space as defined by the user in the PSDsoft Tool. It can also be mapped to Data memory space during Flash memory update or programming.

After reset, the CPU begins execution from location 0000h. As shown in *Figure 9*, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003h. If External Interrupt 0 is going to be used, its service routine must begin at location 0003h. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003h for External Interrupt 0, 000Bh for Timer 0, 0013h for External Interrupt 1, 001Bh for Timer 1 and so forth. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

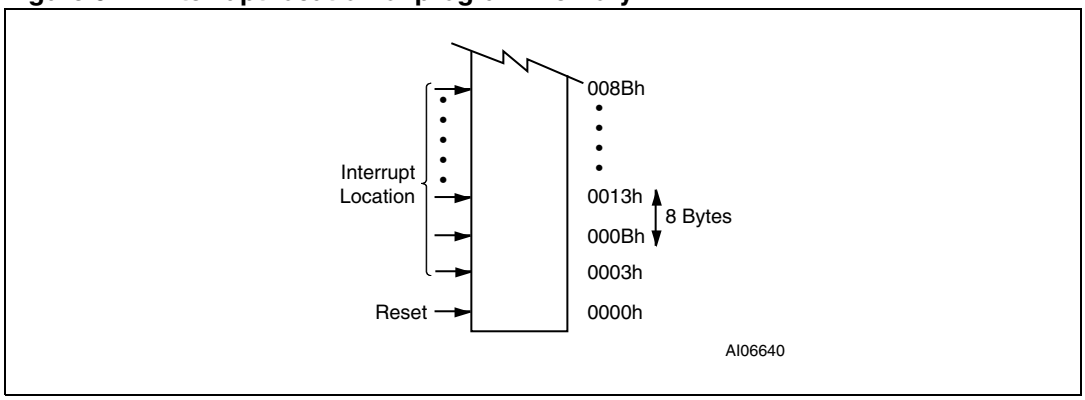
2.4 Data memory

The internal data memory is divided into four physically separated blocks: 256 bytes of internal RAM, 128 bytes of Special Function Registers (SFRs) areas, 256 bytes of external RAM (XRAM-DDC) and 32 Kbytes (XRAM-PSD) in the PSD module.

2.5 RAM

Four register banks, each 8 registers wide, occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack depth is only limited by the available internal RAM space of 256 bytes.

Figure 9. Interrupt location of program memory



2.6 XRAM-DDC

The 256 bytes of XRAM-DDC used to support DDC interface is also available for system usage by indirect addressing through the address pointer DDCADR and data I/O buffer RAMBUF. The address pointer (DDCADR) is equipped with the post increment capability to facilitate the transfer of data in bulk (for details refer to DDC Interface part). However, it is also possible to address the RAM through MOVX command as normally used in the internal RAM extension of 80C51 derivatives. XRAM-DDC FF00 to FFFF is directly addressable as external data memory locations FF00 to FFFF via MOVX-DPTR instruction or via MOVX-Ri instruction. When XRAM-DDC is disabled, the address space FF00 to FFFF can be assigned to other resources.

2.7 XRAM-PSD

The 32 Kbytes of XRAM-PSD resides in the PSD module and can be mapped to any address space through the DPLD (Decoding PLD) as defined by the user in PSDsoft Development tool.

2.8 SFR

The SFRs can only be addressed directly in the address range from 80h to FFh. *Table 15* gives an overview of the Special Function Registers. Sixteen address in the SFRs space are both-byte and bit-addressable. The bit-addressable SFRs are those whose address ends in 0h and 8h. The bit addresses in this area are 80h to FFh.

Table 3. RAM address

Byte address (in hexadecimal)	Bit address (hex)								Byte address (in decimal)
-									-
FFh									255
30h									48
	MSB							LSB	
2Fh	7F	7E	7D	7C	7B	7A	79	78	47
2Eh	77	76	75	74	73	72	71	70	46
2Dh	6F	6E	6D	6C	6B	6A	69	68	45
2Ch	67	66	65	64	63	62	61	60	44
2Bh	5F	5E	5D	5C	5B	5A	59	58	43
2Ah	57	56	55	54	53	52	51	50	42
29h	4F	4E	4D	4C	4B	4A	49	48	41
28h	47	46	45	44	43	42	41	40	40
27h	3F	3E	3D	3C	3B	3A	39	38	39
26h	37	36	35	34	33	32	31	30	38
25h	2F	2E	2D	2C	2B	2A	29	28	37
24h	27	26	25	24	23	22	21	20	36
23h	1F	1E	1D	1C	1B	1A	19	18	35

Table 3. RAM address (continued)

Byte address (in hexadecimal)									Byte address (in decimal)
-									-
22h	17	16	15	14	13	12	11	10	34
21h	0F	0E	0D	0C	0B	0A	09	08	33
20h	07	06	05	04	03	02	01	00	32
1Fh	Register bank 3								31
18h									24
17h	Register bank 2								23
10h									16
0Fh	Register bank 1								15
08h									8
07h	Register bank 0								7
00h									0

2.9 Addressing modes

The addressing modes in UPSD325xx devices instruction set are as follows

1. Direct addressing
2. Indirect addressing
3. Register addressing
4. Register-specific addressing
5. Immediate constants addressing
6. Indexed addressing

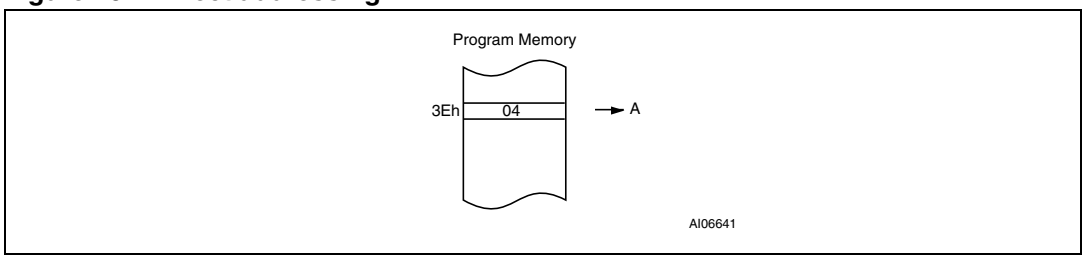
2.9.1 Direct addressing

In a direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs (80~FFH RAM) can be directly addressed.

Example:

mov A, 3EH ; A <----- RAM[3E]

Figure 10. Direct addressing



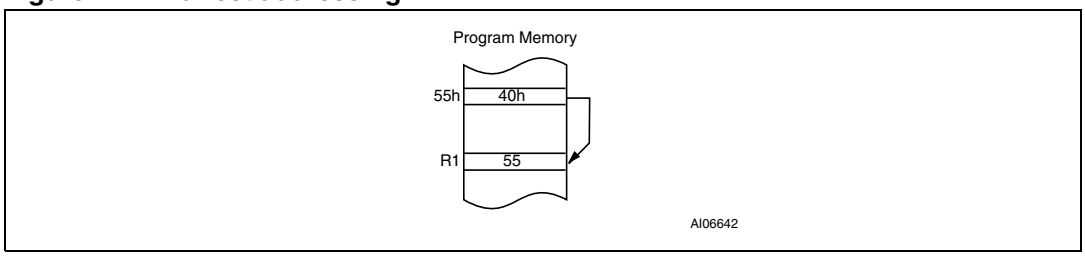
2.9.2 Indirect addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed. The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit “data pointer” register, DPTR.

Example:

```
mov @R1, #40 H ;[R1] <-----40H
```

Figure 11. Indirect addressing



2.9.3 Register addressing

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of four banks is selected at execution time by the two bank select bits in the PSW.

Example:

```
mov PSW, #0001000B ; select Bank0
mov A, #30H
mov R1, A
```

2.9.4 Register-specific addressing

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point it. The opcode itself does that.

2.9.5 Immediate constants addressing

The value of a constant can follow the opcode in Program memory.

Example:

```
mov A, #10H.
```

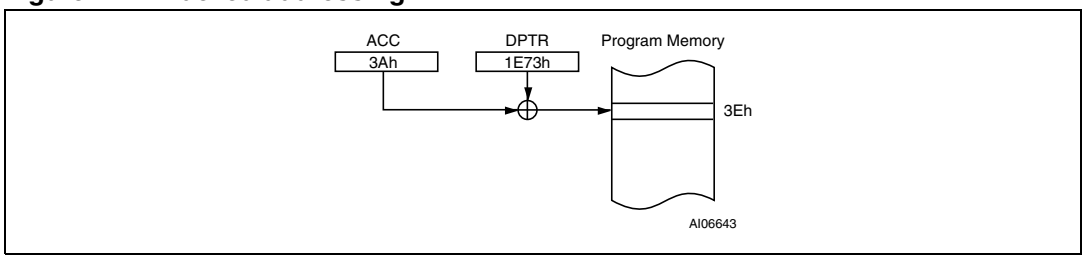
2.9.6 Indexed addressing

Only Program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program memory is formed by adding the Accumulator data to the base pointer.

Example:

movc A, @A+DPTR

Figure 12. Indexed addressing



2.10 Arithmetic instructions

The arithmetic instructions is listed in *Table 4*. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as:

ADD a, 7FH (direct addressing)

ADD A, @R0 (indirect addressing)

ADD a, R7 (register addressing)

ADD A, #127 (immediate constant)

Note: Any byte in the internal Data Memory space can be incremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

In shift operations, dividing a number by 2n shifts its “n” bits to the right. Using DIV AB to perform the division completes the shift in 4?s and leaves the B register holding the bits that were shifted out. The DAA instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DAA operation, to ensure that the result is also in BCD.

Note: DAA will not convert a binary number to BCD. The DAA operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 4. Arithmetic instructions

Mnemonic	Operation	Addressing modes			
		Dir.	Ind.	Reg.	Imm.
ADD A,<byte>	A = A + <byte>	X	X	X	X
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X
SUBB A,<byte>	A = A – <byte> – C	X	X	X	X

Table 4. Arithmetic instructions (continued)

Mnemonic	Operation	Addressing modes			
		Dir.	Ind.	Reg.	Imm.
INC	A = A + 1	Accumulator only			
INC <byte>	<byte> = <byte> + 1	X	X	X	
INC DPTR	DPTR = DPTR + 1	Data Pointer only			
DEC	A = A – 1	Accumulator only			
DEC <byte>	<byte> = <byte> – 1	X	X	X	
MUL AB	B:A = B x A	Accumulator and B only			
DIV AB	A = Int[A / B] B = Mod[A / B]	Accumulator and B only			
DA A	Decimal Adjust	Accumulator only			

2.11 Logical instructions

Table 5 lists logical instructions for UPSD325xx devices. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and byte contains 01010011B, then:

ANL A, <byte>

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in *Table 5*.

The ANL A, <byte> instruction may take any of the forms:

ANL A,7FH(direct addressing)

ANL A, @R1 (indirect addressing)

ANL A,R6 (register addressing)

ANL A,#53H (immediate constant)

Note: Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in:

XRL P1, #0FFH.

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to push it onto the stack in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a

binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

```
MOVE B,#10
DIV AB
SWAP A
ADD A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

Table 5. Logical instructions

Mnemonic	Operation	Addressing modes			
		Dir.	Ind.	Reg.	Imm.
ANL A,<byte>	A = A .AND. <byte>	X	X	X	X
ANL <byte>,A	A = <byte> .AND. A	X			
ANL <byte>,#data	A = <byte> .AND. #data	X			
ORL A,<byte>	A = A .OR. <byte>	X	X	X	X
ORL <byte>,A	A = <byte> .OR. A	X			
ORL <byte>,#data	A = <byte> .OR. #data	X			
XRL A,<byte>	A = A .XOR. <byte>	X	X	X	X
XRL <byte>,A	A = <byte> .XOR. A	X			
XRL <byte>,#data	A = <byte> .XOR. #data	X			
CRL A	A = 00h	Accumulator only			
CPL A	A = .NOT. A	Accumulator only			
RL A	Rotate A Left 1 bit	Accumulator only			
RLC A	Rotate A Left through Carry	Accumulator only			
RR A	Rotate A Right 1 bit	Accumulator only			
RRC A	Rotate A Right through Carry	Accumulator only			
SWAP A	Swap Nibbles in A	Accumulator only			

2.12 Data transfers

2.12.1 Internal RAM

Table 6 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember, the Upper 128 bytes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.