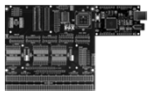
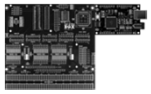


EEL3744

Menu

- What's important?
  - > Architecture Issues
  - > Bus Widths
  - > Non-volatile Memory
  - > Data Memory
- Components of previous processors
- UF Development Board (show & tell)
- Some XMEGA Specs

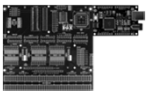
See examples on web-site:
doc8331(XMEGA AU Manual)
doc8385(128A1U Manual)
uPAD Schematic



EEL3744

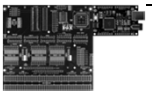
What's important?

- Architectural Issues
  - > Instruction set
  - > Data path widths
  - > Timing and clock cycles
  - > Is buffering required?
  - > Power consumption
  - > Packaging
  - > Timers
  - > I/O ports
  - > A/D, D/A
  - > Power up/down
  - > Battery backup
  - > Family support
  - > Documentation
  - > Part availability

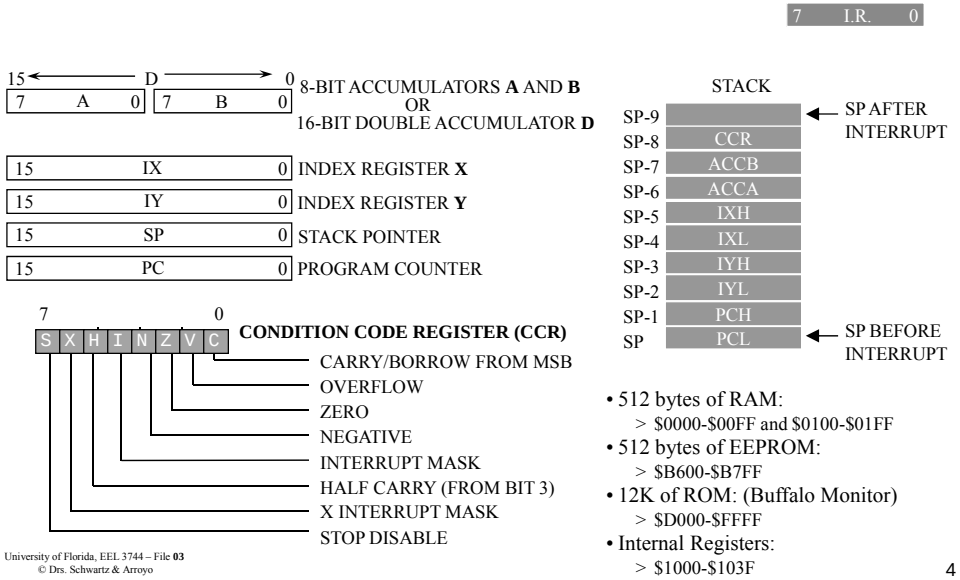


EEL3744 What's important?

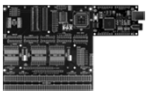
- Bus Widths
  - > Early μP had 4-bit operands.
  - > 68HC11/2 has 8-bit operands and some 16-bit operands. 68HC11 is an 8-bit μC. The 68HC12 is an 8/16-bit μC. Instructions that deal with 16-bit operands take 2 cycles to transfer the 16 bits over the data bus.
  - > The 68HC11/68HC12 (68HC912B32) 16-bit address bus gives access to 2^{16} = 64K addresses.
  - > The TI TMS320F28335 DSC has a 32-bit data bus, but can also be used in a 16-bit data bus mode
  - > **Our XMEGA** is a 8/16-bit processor
    - – 8-bit data
    - – 16-bit addresses registers and 24-bit addresses



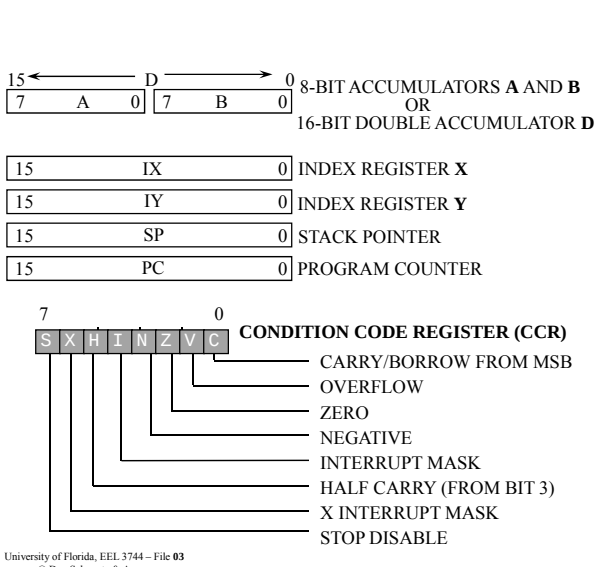
EEL3744 6811 Programming Model

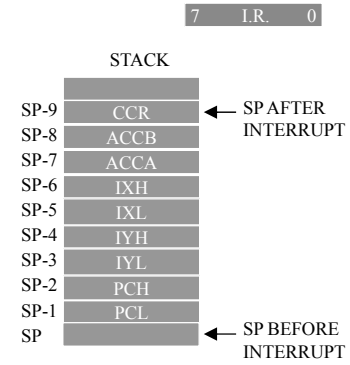


- 512 bytes of RAM:
  - > $0000-$00FF and $0100-$01FF
- 512 bytes of EEPROM:
  - > $B600-$B7FF
- 12K of ROM: (Buffalo Monitor)
  - > $D000-$FFFF
- Internal Registers:
  - > $1000-$103F

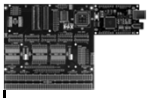


EEL3744 6812 Programming Model





- 512-bytes Internal Registers:
  - > $0000-$00FF and $0100-$01FF
- 1k-bytes Internal SRAM:
  - > $800-$BFF
- 712-bytes Internal EEPROM
  - > $D00-$FFF
- 32k-bytes Internal Flash (single chip) or External Device (expanded)
  - > $8000-$FFFF

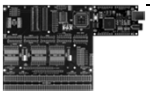


EEL3744

XMEGA CPU General Purpose Working Register Summary

See doc8331 Fig 3-4

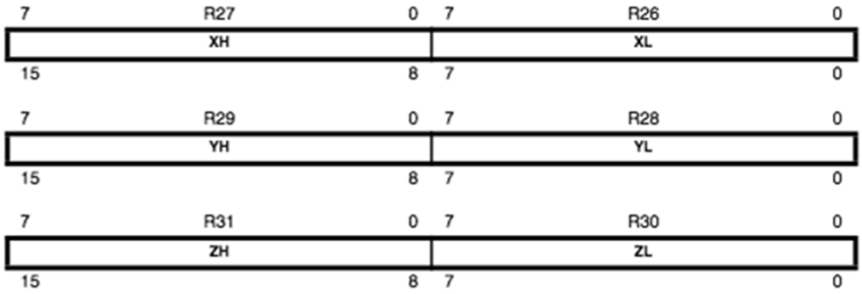
7	0	Addr.
R0		0x00
R1		0x01
R2		0x02
…		
R13		0x0D
R14		0x0E
R15		0x0F
R16		0x10
R17		0x11
…		
X-register Low Byte	R26	0x1A
X-register High Byte	R27	0x1B
Y-register Low Byte	R28	0x1C
Y-register High Byte	R29	0x1D
Z-register Low Byte	R30	0x1E
Z-register High Byte	R31	0x1F

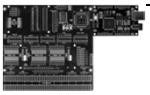


EEL3744 **XMEGA** X, Y, Z Registers

See doc8331 Fig 3-5

- The X, Y, and Z registers can form 16-bit address pointers for addressing of the **Data Memory**
- The Z-register can also be used as an address pointer to read/write to the **Flash Program Memory**, Fuses, Signature Rows, and Lock Bits

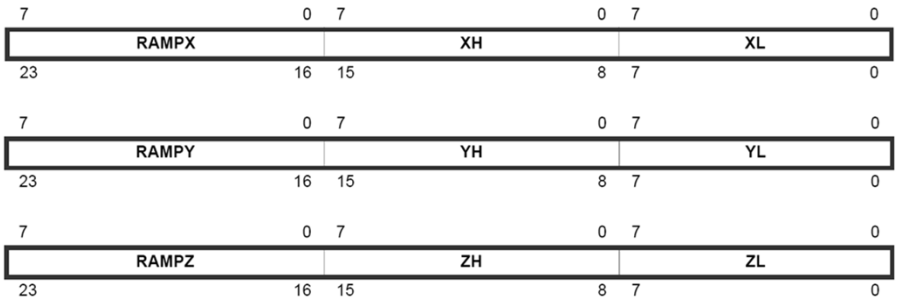


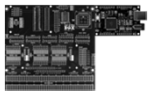


EEL3744 **XMEGA** RAMP_ Extended Indirect Registers

See doc8331 Fig 3-6

- In order to access program memory or data memory above 64KB, the address pointer must be larger than 16 bits
- This is done by concatenating one register to one of the X-, Y-, or Z-registers
  - > This register (RAMPX, RAMPY, or RAMPZ) holds the most-significant byte (MSB) in a 24-bit address or address pointer (2^{24} = 16M, 2^{16} = 64k)



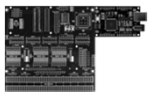


EEL3744

XMEGA Status Register (SREG)

- **SREG**: Status Register
- **C**: Carry Flag
- **Z**: Zero Flag
- **N**: Negative Flag
- **V**: Two's complement overflow indicator
- **S**: N ⊕ V, For signed tests
- **H**: Half Carry Flag
- **T**: Transfer bit used by BLD and BST instructions
- **I**: Global Interrupt Enable/Disable Flag

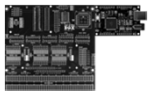
See doc0856 Page 1



EEL3744

XMEGA Status Register (SREG)

- The Status Register contains information about the result of the most recently executed arithmetic or logic instruction
- The Status Register is updated after all ALU operations
  - > This will in many cases remove the need for using the dedicated compare instructions
- The Status Register is **not** automatically stored when entering an interrupt routine nor restored when returning from an interrupt (**unlike** many other uP's)
- The Status Register is accessible in the I/O Memory space

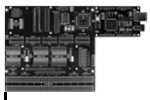


EEL3744

DSC CPU Register Summary [part 1]

See spru430, Table 2-1

Register	Size	Description	Value After Reset
ACC	32 bits	Accumulator	0x00000000
AH	16 bits	High half of ACC	0x0000
AL	16 bits	Low half of ACC	0x0000
XAR0	~~16 bits~~32 bits	Auxiliary register 0	0x00000000
XAR1	32 bits	Auxiliary register 1	0x00000000
XAR2	32 bits	Auxiliary register 2	0x00000000
XAR3	32 bits	Auxiliary register 3	0x00000000
XAR4	32 bits	Auxiliary register 4	0x00000000
XAR5	32 bits	Auxiliary register 5	0x00000000
XAR6	32 bits	Auxiliary register 6	0x00000000
XAR7	32 bits	Auxiliary register 7	0x00000000
AR0	16 bits	Low half of XAR0	0x0000
AR1	16 bits	Low half of XAR1	0x0000
AR2	16 bits	Low half of XAR2	0x0000
AR3	16 bits	Low half of XAR3	0x0000
AR4	16 bits	Low half of XAR4	0x0000
AR5	16 bits	Low half of XAR5	0x0000
AR6	16 bits	Low half of XAR6	0x0000
AR7	16 bits	Low half of XAR7	0x0000

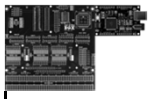


EEL3

DSC CPU Register Summary [part 2]

See spru430, Table 2-1

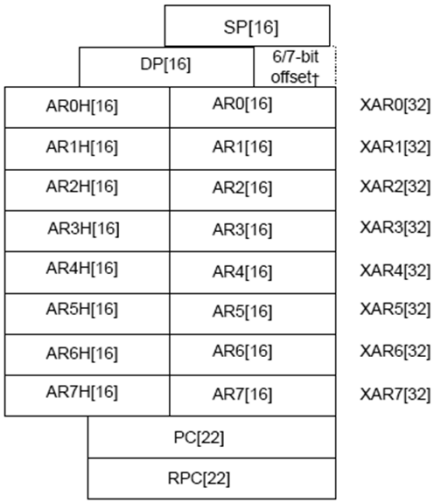
Register	Size	Description	Value After Reset
DP	16 bits	Data-page pointer	0x0000
IFR	16 bits	Interrupt flag register	0x0000
IER	16 bits	Interrupt enable register	0x0000 (INT1 to INT14, DLOGINT, RTOSINT disabled)
DBGIER	16 bits	Debug interrupt enable register	0x0000 (INT1 to INT14, DLOGINT, RTOSINT disabled)
P	32 bits	Product register	0x00000000
PH	16 bits	High half of P	0x0000
PL	16 bits	Low half of P	0x0000
PC	22 bits	Program counter	0x3F FFC0
RPC	22 bits	Return program counter	0x00000000
SP	16 bits	Stack pointer	0x0400
ST0	16 bits	Status register 0	0x0000
ST1	16 bits	Status register 1	0x080B†
XT	32 bits	Multiplicand register	0x00000000
T	16 bits	High half of XT	0x0000
TL	16 bits	Low half of XT	0x0000

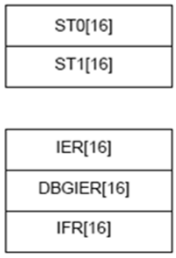


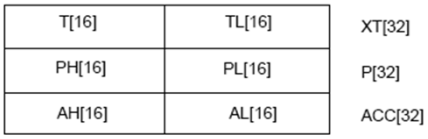
EEL3744

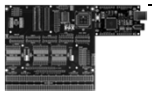
DSC CPU Registers

See spru430, Fig 2-2

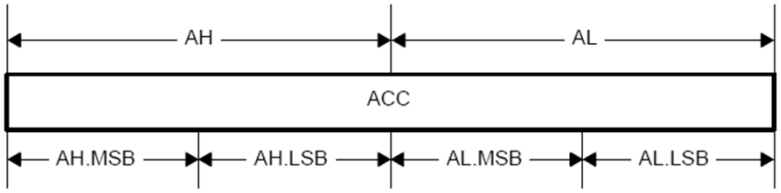








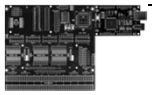
EEL3744 **DSC** Accumulator (ACC)



- ACC (32 bits) = AH (16 bits) | AL (16 bits)
  - > The high or low bytes of AH or AL can be accessed independently
    - – AH.MSB, AH.LSB, AL.MSB, AL.LSB

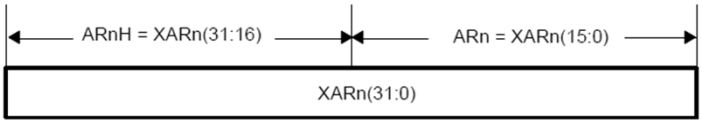
See spru430, Fig 2-3

- Main working register
- Destination for all ALU operations except those that operate directly on memory or registers
- Associated status bits (in ST0 status register): C, Z, N, V, etc.



EEL3744 **DSC** Auxiliary Registers (XAR0−XAR7, AR0−AR7)

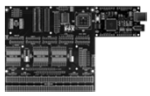
- XAR0-XAR7 are used as pointers to memory or as general-purpose registers



See spru430, Fig 2-8

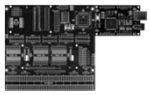
n = number 0 through 7

- ARx are 16-bit registers are the low 16-bits
  - AR0 = XAR0(15:0) and AR7 = XAR7(15:0)
- ARHx are 16-bit registers are the high16-bits
  - ARH0 = XAR0(31:16)
  - ARH7 = XAR7(31:16)



EEL3744 **DSC** Program Counter (PC)

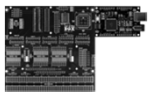
- Program Counter points to the instruction that is currently being processed
- PC is 22-bits, i.e., PC(21:0)



EEL3744

XMEGA A1U Specs

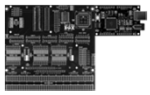
- We have a **ATxmega128A1U**
- Nonvolatile program and data memories
  - > 128KBytes of in-system self-programmable flash
  - > 8KBytes boot section
  - > 2KBytes EEPROM
  - > 8KBytes internal SRAM
- External bus interface for up to 16Mbytes SRAM
  - > $2^{24} = 2^4 * 2^{20} = 16 * 1M = 16M$
- External bus interface for up to 128Mbit SDRAM



EEL3744

XMEGA A1U Specs

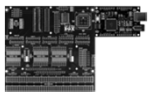
- Peripheral features
  - > Four-channel DMA controller
    - – All peripherals can use DMA for data transfer
  - > Eight-channel event system
- Eight 16-bit timer/counters
  - > Four timer/counters with 4 output compare or input capture channels
  - > Four timer/counters with 2 output compare or input capture channels
  - > High resolution extension on all timer/counters
  - > Advanced waveform extension (AWeX) on two timer/counters



EEL3744

XMEGA A1U Specs

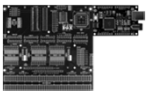
- One USB device interface (31 endpoints)
  - > USB 2.0 full speed (12Mbps) and low speed (1.5Mbps)
  - > 32 Endpoints with full configuration flexibility
- Eight USARTs with IrDA support for one USART
- Four two-wire interfaces with dual address match (I2C and SMBus compatible)
- Four serial peripheral interfaces (SPIs), i.e., synchronous serial
- 16-bit real time counter (RTC) with separate oscillator



EEL3744

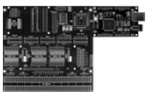
XMEGA A1U Specs

- Two sixteen channel, 12-bit, 2msps ADCs
  - > msps = million samples/sec
- Two two-channel, 12-bit, 1msps DACs
- Four Analog Comparators (ACs) with window compare function, and current sources
- Touch sensing interfaces: button, sliders & wheels
  - > QTouch® library support
    - – Capacitive touch buttons, sliders and wheels
- 32 PWM outputs, 8 UART, 4 TWI (I2C) and 4 SPI channels, and a CRC generator module.



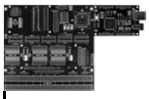
EEL3744 **XMEGA** A1U Specs

- External interrupts on all general purpose I/O pins
- Programmable watchdog timer with separate on-chip ultra low power oscillator
- Special microcontroller features
  - > Power-on reset and programmable brown-out detection
  - > Internal and external clock options with PLL and prescaler
  - > Programmable multilevel interrupt controller
  - > Five sleep modes
- Programming and debug interfaces
  - > JTAG (IEEE 1149.1 compliant) interface, including boundary scan
  - > PDI (Program and Debug Interface)



EEL3744 **XMEGA** A1U Specs

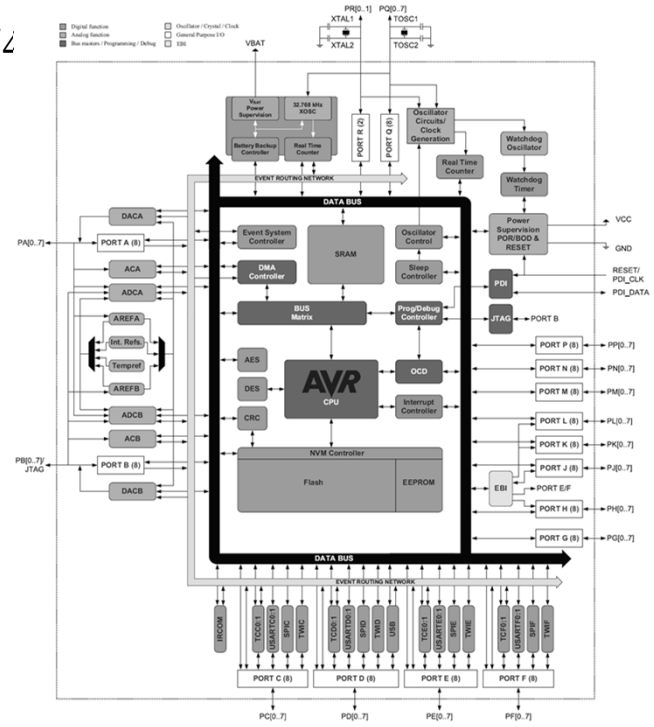
- I/O and packages
  - > 78 Programmable I/O pins
  - > 100 lead TQFP (**we have this!**) [TQFP = Thin Quad Flat Pack]
  - > 100 ball BGA [BGA = Ball Grid Array]
  - > 100 ball VFBGA [VFBGA = Very Fine-Pitch Ball Grid Array]
- Operating voltage
  - > 1.6 – 3.6V
- Operating frequency
  - > 0 – 12MHz from 1.6V
  - > 0 – 32MHz from 2.7V

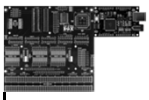


EEL37

XMEGA Block Diagram

See doc8331 Fig 2-1

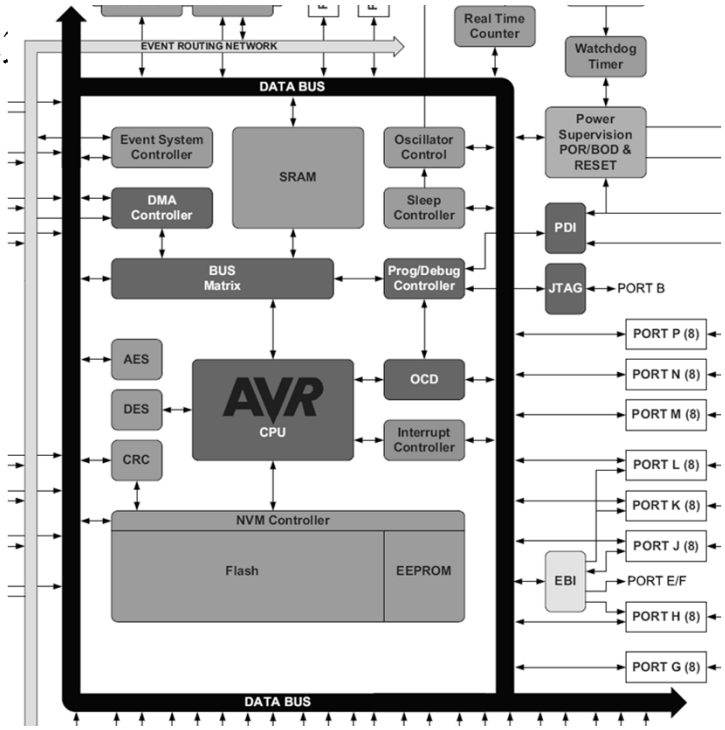


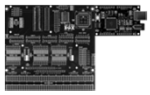


EEL

XMEGA Block Diagram

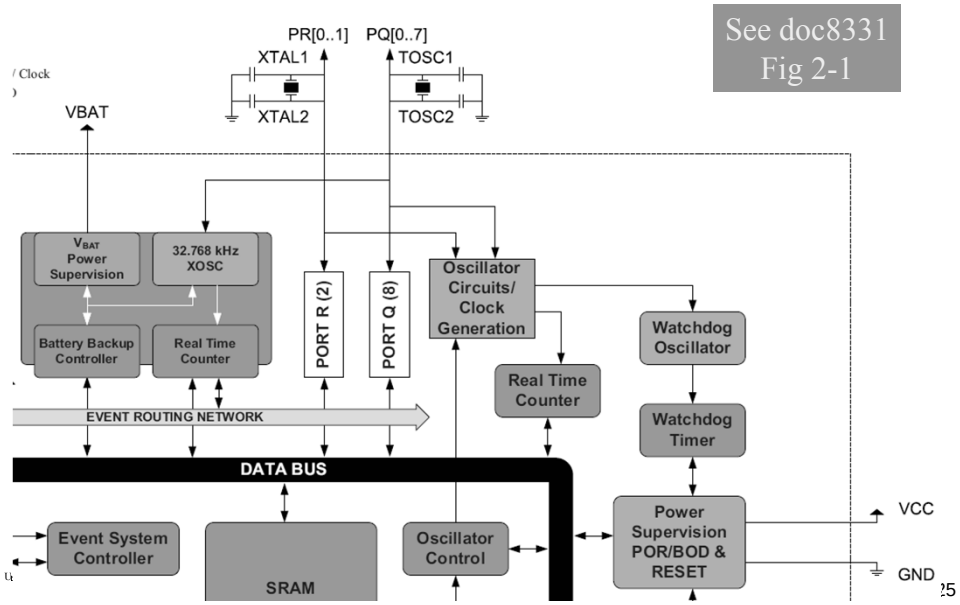
See doc8331 Fig 2-1



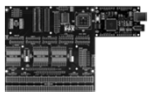


EEL3744 **XMEGA** Block Diagram

See doc8331 Fig 2-1

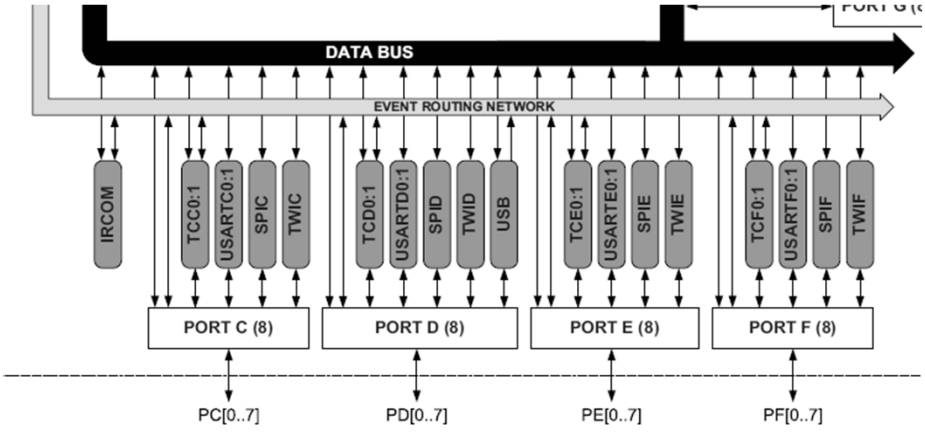


25

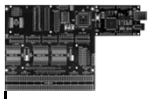


EEL3744 **XMEGA** Block Diagram

See doc8331 Fig 2-1



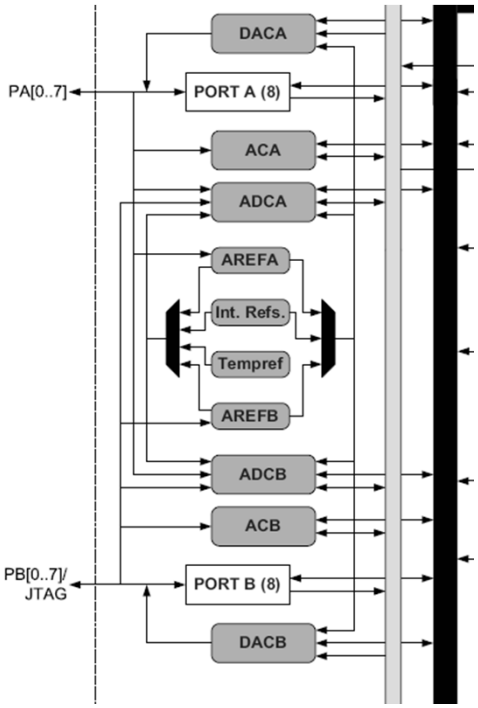
26

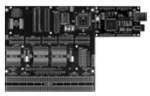


EEL3744

XMEGA Block Diagram

See doc8331 Fig 2-1

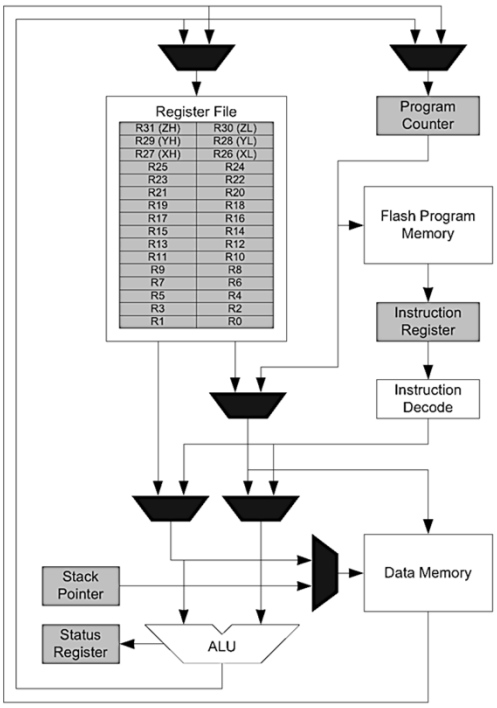


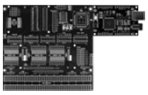


EEL3744

Conceptual **XMEGA** CPU Block Diagram

See doc8331 Fig 3-1

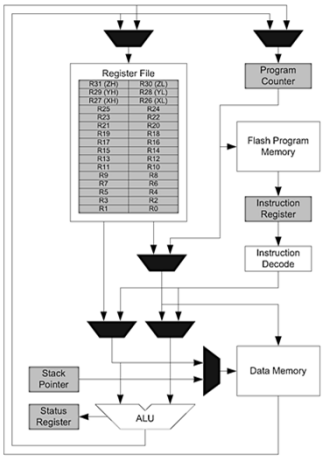




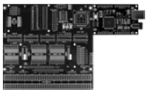
EEL3744

XMEGA Accumulator

- All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU)
  - > This allows **two independent registers** to be accessed in **one single instruction**, and executed in **one clock cycle**.
- Resulting architecture is more code efficient and faster than conventional single-accumulator or CISC based microcontrollers



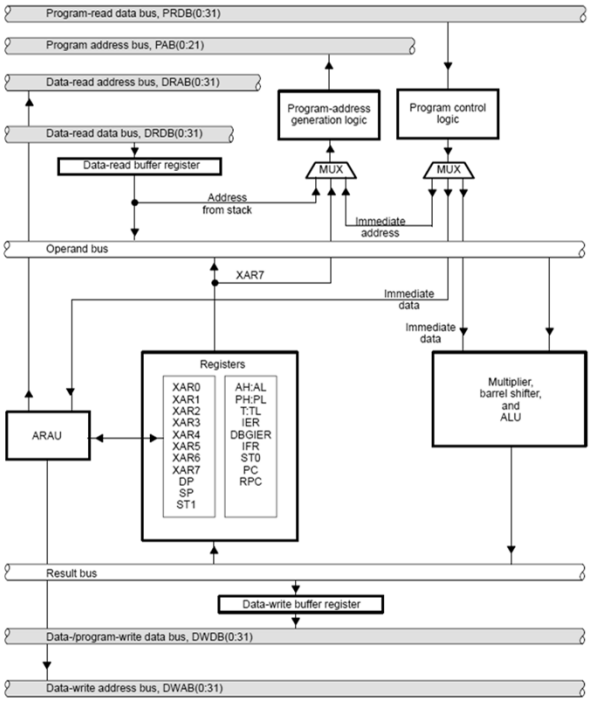
See doc8331 Fig 3-1

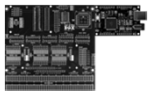


EEL3744

Conceptual **DSC** CPU Block Diagram

See spru430, Figure 2-1

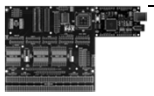




EEL3744

XMEGA is Little Endian

- When 32-bit values are saved the least significant 16 bits are saved first, in the lower address (and the most significant 16 bits are saved to the next higher address), i.e., XMEGA is a **little endian** processor
- Loading a single byte of the address register will always update the LSB byte while the MSB bytes are left unchanged
  > See example



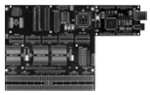
EEL3744

XMEGA Flash **Program** Memory

- Our XMEGA contain on-chip, in-system re**program**mable flash **memory** for program storage
- The flash memory can be accessed for read and write from an external programmer
- All AVR CPU instructions are 16 bits wide, and each flash location is **16 bits wide**
- **Program Memory Map:**

See doc8385 Fig 7-1

Word Address (hex)	# Addresses	Description
0 – EFFF	60k	Application Section, 60k (120K)
F000 – FFFF	4k	Application Table Section, 4k (8K)
1 0000 – 1 0FFF	4k	Boot Section, 4k (8K)



EEL3744

Data Memory Map for **XMEGA**

- Our XMEGA **data memory** contains the I/O memory, internal SRAM, optionally memory mapped EEPROM, and external memory, if available
  - > **EEPROM:** It is either addressable in a separate data space (default) or memory mapped and accessed in normal data space
    - – Memory mapped EEPROM starts at 0x1000
  - > **Data Memory Map:**

Addr (hex)	Description
0 – 0FFF	I/O Registers (4kB)
1000 – 17FF	EEPROM (2kB)
1800 – 1FFF	Reserved
2000 – 3FFF	Internal SRAM (8kB)
4000 – FF FFFF	External Memory (0 to ~16MB)

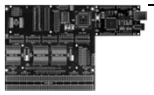
See doc8385 Fig 7-2

EEL 3744

See doc8385 Section 7.5-7.8

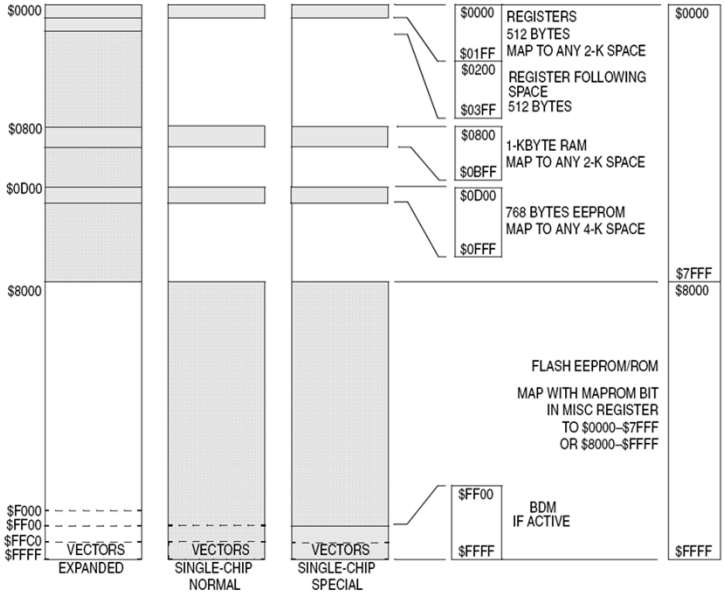
Data Memory Map for **XMEGA**

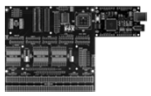
- > **I/O Memory:** The status & configuration registers for peripherals and modules, including the CPU, are addressable through I/O memory locations
  - – All I/O locations can be accessed by the load (**LD/LDS/LDD**) and store (**ST/STS/STD**) instructions, which are used to transfer data between the **32 registers** in the register file and the I/O memory
  - – The **IN** and **OUT** instructions can address I/O memory locations in the range **0x00 - 0x3F** directly (**GPIO**, **VPORT0-3**, **CPU**)
    - ▪ See doc8385, Table 34-1, for Peripheral Module Address Map
- > **General Purpose I/O Registers:** The lowest 16 I/O memory addresses (**R0-R15**) are reserved as general purpose I/O registers
  - – These registers can be used for storing global variables and flags, as they are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions
- > **External Memory:** Four ports can be used for external memory, supporting external SRAM, SDRAM, and memory mapped peripherals such as LCD displays (see “EBI – External Bus Interface” in section 28)



EEL3744 Tech: Fig 5-6: Memory Map for **68HC12** (68HC912B32)

- Same memory space is used for **BOTH** program and data memory

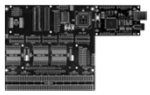




EEL3744 Non-volatile Memory: ROM

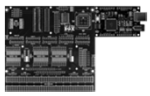
- Non-volatile Memory - memory which holds its contents when power to the chip is removed.
  - > ROM (Read Only Memory) is one example

    From ~1990
    - – One time mask charge of $3k-$4k and minimum of 1000 pieces is typical. Lag time is usually about 10 weeks.
    - – 1000 pieces: If $14/part then $4k + $14×1000 = $18k
    - – 100,000 pieces: If $7/part then $4k + $7×100000 = $704k (39 times more than 1000 pieces)
    - – During debugging, CAN NOT use ROM version of μP. Use emulator or programmable version.



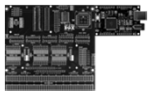
EEL3744
Non-volatile Memory: ROM

- Non-volatile Memory (cont.)
  - > EPROM (**E**rasable, **P**rogrammable **R**ead **O**nly **M**emory)
    - – Many older μC's have this
  - > EEPROM (**E**lectrically **E**rasable, **P**rogrammable **R**ead **O**nly **M**emory) or Flash (a type of EEPROM):
    - – All modern μC's have this
- Data Memory – Most μC's have two or three basic types.
  - > ROM: for programs (operating system)
  - > RAM: for variables
  - > EEPROM and/or Flash: for non-volatile storage of constant or variables



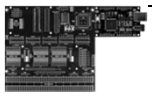
EEL3744 **XMEGA** Non-Volatile Memory (NVM)

- XMEGA has EEPROM memory for non-volatile data storage
  - > It is addressable either in as a separate data space (default), or it can be memory mapped and accessed in normal data space.
- The EEPROM memory supports both byte and page access
- NVM Controller includes
  - > Memory Address Registers (ADDR0,ADDR1,ADDR2)
  - > Memory Data Registers (DATA0,DATA1,DATA2)
  - > Memory Command Register (CMD)
  - > Memory Control Registers (CTRLA,CTRLB)
  - > Interrupt Control Register (INTCTRL)
  - > Memory Status Register (STATUS)
  - > Memory Lock Bit Register (LOCKBITS)



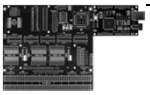
EEL3744 Motorola/Freescale 68HC**11**E9 Specs

- 12K ROM/EPROM
- 512 bytes RAM (0.5μs read/write)
- 512 bytes EEPROM
  - > Can erase 1 byte, 1 row (of 16 bytes), or erase all.
  - > Programs (or writes) 1 byte at a time.
- 5 I/O ports (A-E)
  - > Some are pins are bi-directional (programmable)
  - > Input pins (input to CPU) READ
  - > Output pins (out of the CPU) WRITE
- 16-bit address space (and address bus).
- 8 MHz clock speed (but bus speed of 2 MHz, the so-called E-clock)



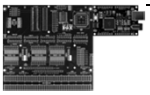
EEL3744 Motorola/Freescale 68HC**11**E9 Specs

- Timer (16-bit with prescaler) (See RM: page 10-8)
  - > Resolution of from 500 ns = 0.5 μs to 8 μs.
  - > Range of 32.77 ms up to range of 524.3 ms.
  - > Up to 4 (bit) input capture functions
  - > Up to 5 (bit) output compare functions
- Real-Time Interrupt
- 8 channel, 8-bit A/D
- Asynchronous SCI (Serial Communications Interface)
- Synchronous SPI (Serial Peripheral Interface)
- 8-bit Pulse Accumulator
- Some software features:
  - > 8-bit multiply
  - > 16-bit divide
  - > Bit manipulation
  - > Wait and Stop modes



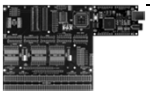
EEL3744 Motorola/Freescale **68HC12** (68HC912B32) Specs

- **1K** RAM
- **768** bytes EEPROM
  - > Can erase 1 byte, 1 row (of 32 bytes), or erase all.
  - > Programs (or writes) 1 byte at a time.
- **32K FLASH** (in single-chip mode)
- **8 I/O ports (A, B, E, AD, S, T, P, DLC)**
  - > Some are pins are bi-directional (programmable)
  - > Input pins (input to CPU) READ
  - > Output pins (out of the CPU) WRITE
- 16-bit address space (and address bus).
- Up to **16 MHz** clock speed (but bus speed is ½ that, the so-called E-clock)
  - > We'll use **4 MHz** clock speed and E = 2 MHz



EEL3744 Motorola/Freescale **68HC12** (68HC912B32) Specs

- Timer (16-bit with prescaler)
  - > **Resolution of from 500 ns = 0.5 μs = (1/E) to 16 μs.**
  - > **Range of 32.768 ms up to range of 1.049 s.**
  - > Up to 8 (bit) input capture functions
  - > Up to 8 (bit) output compare functions
- Real-Time Interrupt
- 8 channel, 8-bit A/D
- Asynchronous SCI (**S**erial **C**ommunications **I**nterface)
- Synchronous SPI (**S**erial **P**eripheral **I**nterface)
- 8-bit Pulse Accumulator
- **Instruction Queue**
- Some software features:
  - > **8-bit & 16-bit multiplies, 16-bit & 32-bit divides**
  - > Bit manipulation
  - > **Fuzzy Logic instructions**
  - > Wait and Stop modes

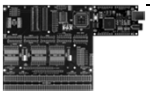


EEL3744 Some TI TMS320F28335 **DSC** Specs

- 34k x 16 SRAM
- 256k x 16 Flash ROM
- 8k x 16 Boot ROM

See sprs439, Sec 1.1

- 16-bit or 32-bit external interface bus (2M x 16)
- 16x16 and 32x32 MAC (multiply and accumulate)
  > 16x16 Dual MAC
- Harvard Bus Architecture
- Code Efficient for C/C++ and Assembly
- Up to **150 MHz** clock speed
  > We used 15 MHz clock speed (7.5 MHz external speed)
  > On chip oscillator
  > PLL for dynamically changing clock speed
  > Watchdog timer

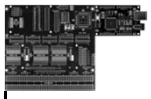


EEL3744 TI TMS320F28335 **DSC** Specs

- 6-channel DMA (Direct Memory Access)
- Three 32-bit Timers
  > 18 PWM Outputs (max)
  > 6 Input Captures (max)
  > 2 Quadrature Encoder Interfaces (max)
  > 8 32-bit or 9 64-bit timers (max)

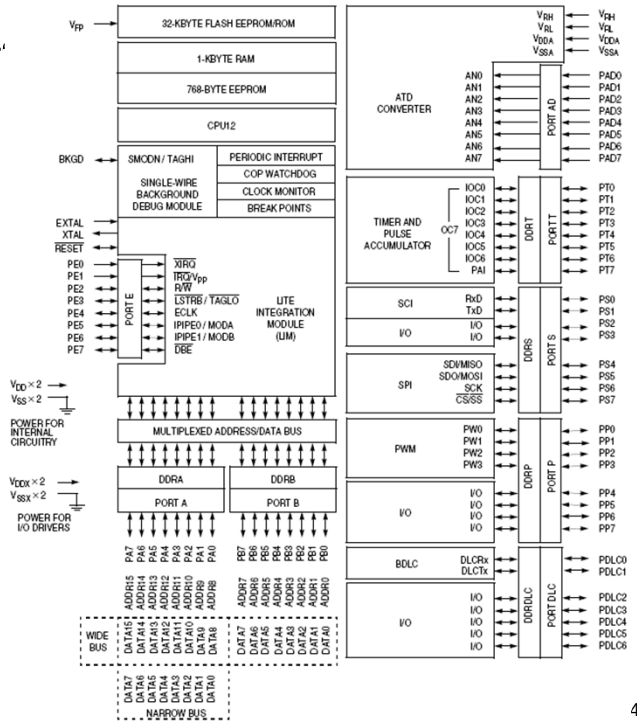
See sprs439, Sec 1.1

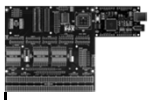
- Serial Ports
  > 3 SCI [UART] (max)
  > 1 SPI, 1 I2C, 2 CAN (max)
  > 1 McBSP (Multichannel Buffered Serial Port)
- 12-bit 16-channel A-to-D
- 88 GPIO (max),
  > Individually programmable as input or output



EEL374

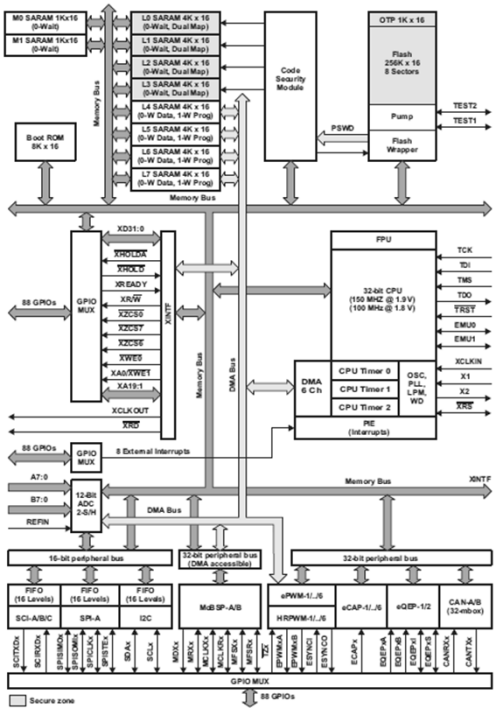
Tech: Fig.1-1:
68HC12
(68HC912B32)
Block Diagram

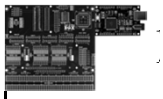




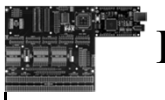
EEL3744

TI **DSC**
Functional Block
Diagram
(spru439, Fig 3-1)

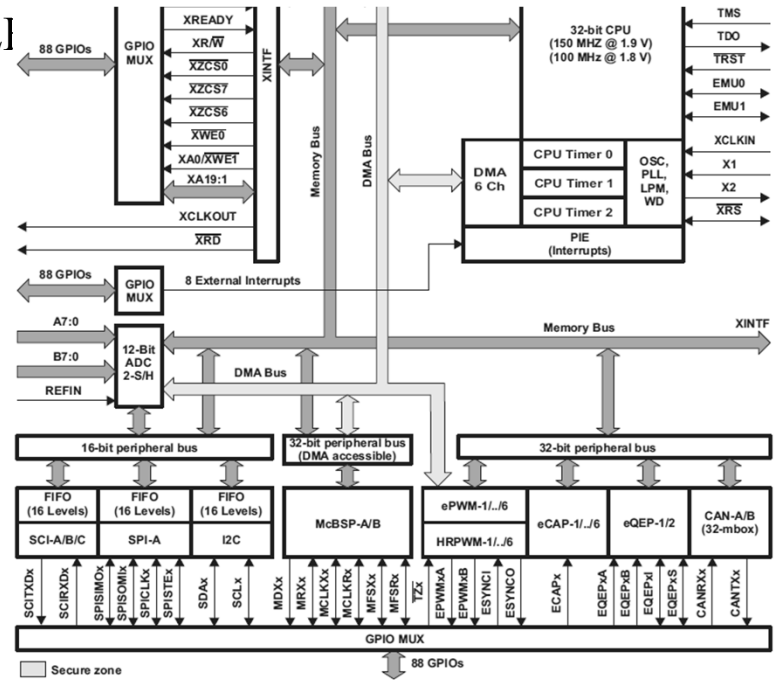


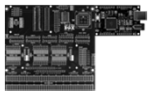


TI **DSC** Functional Block Diagram (spru439, Fig 3-1) [top of figure]



TI **DSC** Functional Block Diagram (spru439, Fig 3-1) [bottom of figure]

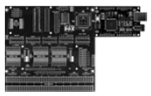




EEL3744

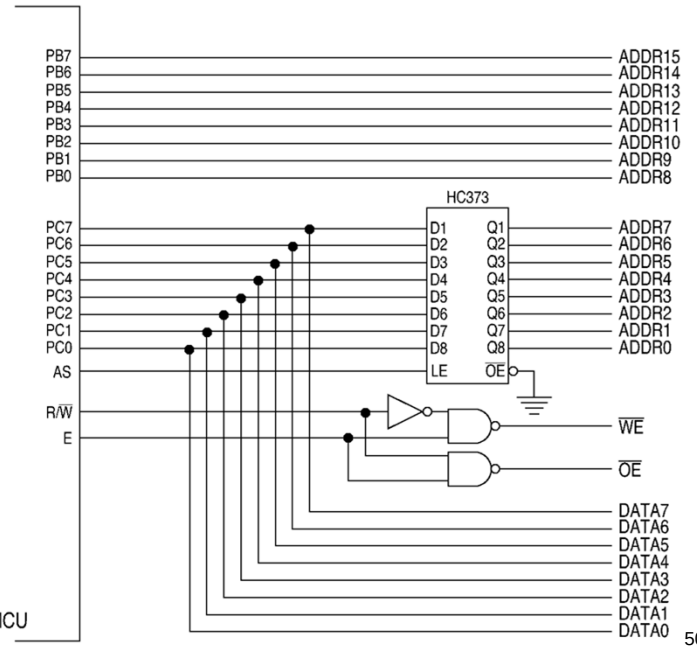
Shared address/data bus

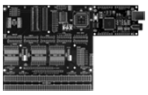
- Many μC's and μP's (especially older ones) have shared address and data bus
  - > Many that do not have a shared bus option do **NOT** have external busses at all!
  - > 68HC11 uses 8-pins, AD7-0 for 8-bits of data and 8-bits of address
  - > For the 68HC12, you must latch all 16-pins of the address/data bus, even if you are only using 8-bits of data



EEL3744

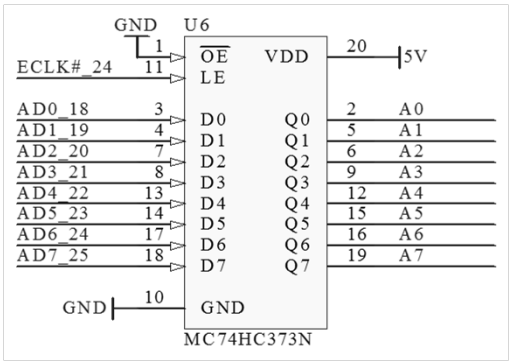
68HC11 Address/Data Demultiplexing

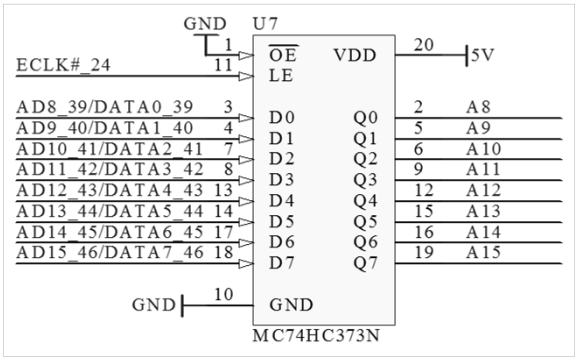


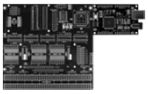


EEL3744

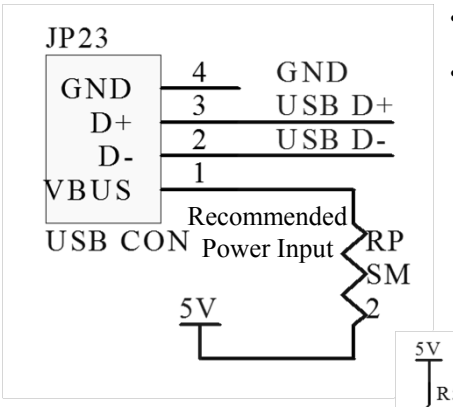
UF **68HC12**-Board Schematic: Address/Data Demultiplexing



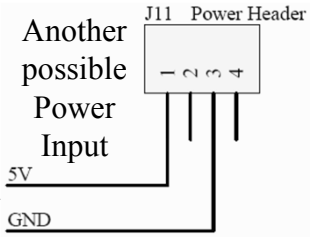


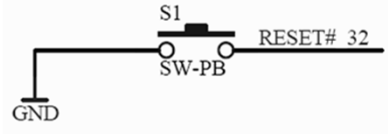


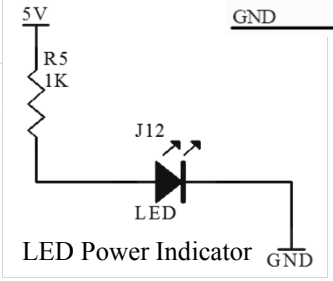
EEL3744 UF **68HC12** Board Schematic: Power from USB and Reset Circuits

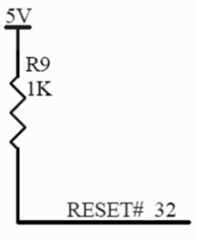


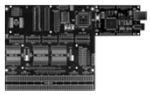
- If Power comes from USB, then **install** RP solder jumper
- If Power to come from external regulated supply, then do **NOT** install RP solder jumper







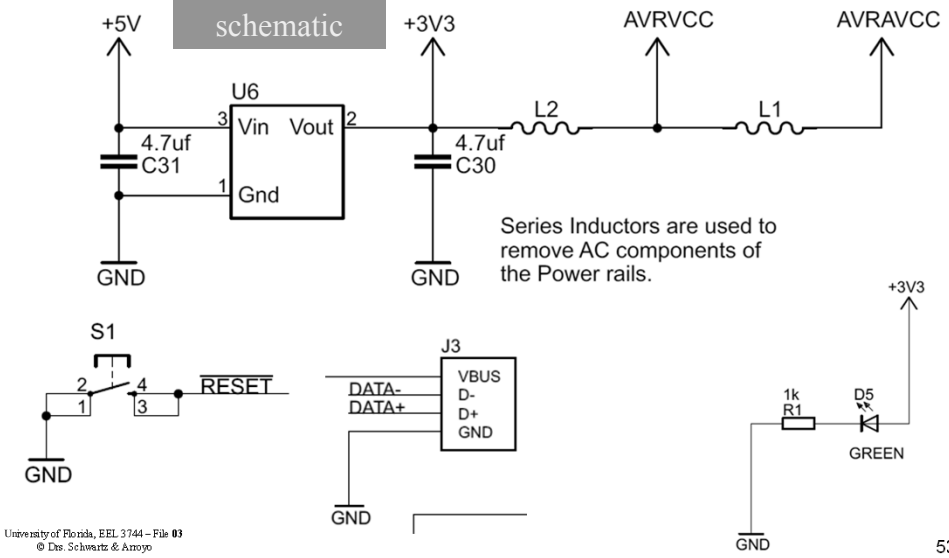


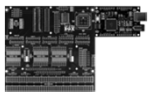


EEL3744

UF **uTinkerer** Board Schematic: Power from USB and Reset Circuits

See uTinkerer schematic





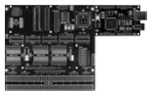
EEL3744

UF **uPAD** Board Schematic: Power from USB and Reset Circuits

See uPAD schematic for **power circuits**, pages 3-4

See also Proto Base schematic

http://mil.ufl.edu/3744/docs/uPAD/


EEL3744

The End!