ESD 1 Exam 1 Cheat Sheet

Tuesday, October 1, 2024 17:52

E.S.

- Specific application that is known at design time.
- Not programmable by end user
- Optimized fixed runtime requirements.

PC

- Broad class of applications.
- Programmable by end user.
- Faster is better.

Assembly

Specific to a given processor.

Mnemonics = names of low level machine instructions.

Labels = names of variables or mem addresses.

Directives: define data and constants.

Macros: facilitate the inline expansion of text into other code.

Registers

32 total

R0 always holds 0

E.S. Components

- Processor
- Memory
- I.O.

E.S. Tech

ASIC - Application Specific IC
- fixed in silicon

ASSP - Application Specific Standard Parts
- fixed in silicon

SOC - System on Chip
- fixed in Silicon

SOPC - System on Programmable Chip

FPGA - not fixed in Silicon.

CISC

- Complex instr. set computer
- Emphasis on HW, simple SW
- Multiclock, complex instr
- Mem-to-mem, load and Store incorporated into instr.
- Small code sizes, high Cycles/second.
- Transistors used for storing complex instr.

RISC

- Reduced Instr. set Computer.
- Emphasis on SW, simple HW.
- Single clock.
- Register-to-Register, load and store are independent to instr.
- Large code, low cycles/sec.
- Spends more transistors on mem. registers.

Cores

IP - Intellectual property
- Ready made function that you can instantiate in your design.

Soft processor - a processor fully described in software, usually in an HDL, which can be synthesized in programmable hardware.

Hard processor - processor is embedded in the silicon of the FPGA.

C Review

Scope: Part of a program within which a name can be used.

Visibility: Region of program text from which the object associated with the name can be legally accessed.

Typedef: Creates custom datatypes.

Preprocessor: part of compiler.
- #include, #ifndef, #define, etc.

Type Qualifiers:

Volatile: Can change beyond control of program.

Polling vs Int.

Polling: programmer controls I/O device access.

Int: Device has ability to alert processor when servicing is needed.

ISR:

- Int. handler.
- Should be kept short.
- Each ISR handle one specific device.
- first should verify source of int.
- last should be to clear int. event.

Type Qualifiers:
Volatile: Can change beyond control of program.
Restrict: Pointer it is qualifying is the only direct or non direct way of accessing.

Storage classes:
- auto
- register
- extern
- static

Bitwise Ops:
& - and
| - or
^ - xor
<< - shift left
>> - shift right
~ - 1's complement

Interrupts:
level: intr. sig. by steady state of line (1 or 0)
As long as signal stays asserted an int. would be issued.
Edge: int. signaled by a level transition on line.

- first should verify source of int.
- last should be to clear int. event.

Altera Avalon Interface:
- provides easy connection to components.
- use interface for custom component.
- Component can include any # of Avalon interfaces.
- Avalon interfaces are open standard.
- Each A.i. defines its own signals and behavior.

Avalon Conduits:
- used to group together an arbitrary collection of signals.
- used to route signals out of / into QSYS system.

I/O Devices:
- provide/receive info to/from processor.

- Uint32 *ptr32 = 0x0018; // uin32 is 4 bytes
- *(ptr32 + 8) = 0x00000004; // Increment address 8 times by 4 .(0 through 8)
- //0x0018 + 1 = 0x0018 + 0x0022
- // 18 = 24 in decimal
- // 24 + 32 = 56
- // 56 = 38 in hex
- // 0x0038;
- // 0x00000004 is being stored in 0x0038 memory location.

```
#define BASE_ADDRESS 0x12345678
uint16* my_device = (uint16*)BASE_ADDRESS
*(my_device+2) |= 0x02;  /* Enable interrupt in bit 2 of key interrupt mask register */
```