* Date: 2/21/2014
* To: CSS 422, Winter 2014
* From: Will Tanna, Team Leader
* Subject: Team Bits Third Group Project Report

**Work Completed:**

1. Group read Clements chapters for better understanding of Assembly
2. Reworked the code of our I/O to include subroutines
3. Reserved a room in the library for Saturday 2/22/14 to work on the project as a group for 8 hours
4. Created new easy to ready Opcode table in Excel (page 2 and 3)
5. Walked through our current I/O code to find out what the next step was
6. Started creating I/O code to send hex code of instruction to op

**Problems:**

Reading the Clements chapters has taught us a lot about different ways to do the same thing in Assembly. We originally just had one loop for the I/O where all the code was going to be worked on, but once we learned subroutines and JMP tables we found that our previous code was not going to be well optimized or really work for what we want to do.

**Work Scheduled:**

1. All meet together and work on the project for 8 hours
2. Get I/O to send data to OP by Saturday
3. Get OP code finished for MOVE by this weekend if not early next week
4. Start EA code

**Self Evaluation:**

Our group has put a lot of time and effort from the start. We have spent a lot of time reading and analyzing assembly code. Although we are still confused on a lot of things, we feel like we have a great head start into this project and are really trying to get this perfect and complete as soon as possible.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mnemonic | Size | | | Single Effective Address Operation Word | | | | | | | | | | | | | |  |  | Notes |
| ADD | B | W | L | 1 | 1 | 0 | 1 | Dn | | | D | S | | M | | | Xn | | |  |
| ADDA |  |  | L | 0 | 1 | 1 | 1 | An | | | S | 1 | 1 | M | | | Xn | | |  |
| ADDI |  | W | L | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | S | | M | | | Xn | | |  |
| AND | B | W | L | 1 | 1 | 0 | 1 | Dn | | | D | S | | M | | | Xn | | |  |
| ANDI |  | W | L | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | S | | M | | | Xn | | |  |
| ASR | B | W | L | 1 | 1 | 1 | 0 | COUNT | | | dr | S | | ir | 0 | 0 | Xn | | | Register Shifts |
| ASR | B | W | L | 1 | 1 | 1 | 0 | 0 | 0 | 0 | dr | 1 | 1 | M | | | Xn | | | Memory Shifts |
| BCHG |  | W | L | 0 | 0 | 0 | 0 | Dn | | | 1 | 0 | 1 | M | | | Xn | | |  |
| BGT | B | W | L | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | Displacement | | | | | | | | Bcc |
| BLE |  | W |  | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | Displacement | | | | | | | | Bcc |
| BVS |  | W |  | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Displacement | | | | | | | | Bcc |
| CLR |  | L |  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | S | | M | | | Xn | | |  |
| CMP | B | W | L | 1 | 0 | 1 | 1 | Dn | | | 0 | S | | M | | | Xn | | |  |
| CMPA | B | W | L | 1 | 0 | 1 | 1 | An | | | S | 1 | 1 | M | | | Xn | | |  |
| CMPI | B | W | L | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | S | | M | | | Xn | | |  |
| DIVU | B | W | L | 1 | 0 | 0 | 0 | Dn | | | 0 | 1 | 1 | M | | | Xn | | |  |
| EOR | B | W | L | 1 | 0 | 1 | 1 | Dn | | | 1 | S | | M | | | Xn | | |  |
| EORI | B | W | L | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | S | | M | | | Xn | | |  |
| JSR | B | W | L | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | M | | | Xn | | |  |
| LEA | B | W | L | 0 | 1 | 0 | 0 | An | | | 1 | 1 | 1 | M | | | Xn | | |  |
| LSL | B | W | L | 1 | 1 | 1 | 0 | COUNT | | | dr | S | | ir | 0 | 1 | Xn | | | Register Shifts |
| LSL | B | W | L | 1 | 1 | 1 | 0 | 0 | 0 | 1 | dr | 1 | 1 | M | | | Xn | | | Memory Shifts |
| MOVE | B |  | L | 0 | 0 | S |  | Xn | | | M | | | M | | | Xn | | |  |
| MOVEM | B | W | L | 0 | 1 | 0 | 0 | 1 | dr | 0 | 0 | 1 | S | M | | | Xn | | |  |
| MOVEQ |  | W | L | 0 | 1 | 1 | 1 | Dn | | | 0 | Data | | | | | | | |  |
| MULS | B | W | L | 1 | 1 | 0 | 0 | Dn | | | 1 | 1 | 1 | M | | | Xn | | |  |
| ROL | B | W | L | 1 | 1 | 1 | 0 | COUNT | | | dr | S | | ir | 1 | 1 | Dn | | | Register Rotate |
| ROL, ROR | B | W | L | 1 | 1 | 1 | 0 | 0 | 1 | 1 | dr | 1 | 1 | M | | | Xn | | | Memory Rotate |
| RTS | B | W | L | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |  |
| SUB | B | W | L | 1 | 0 | 0 | 1 | Dn | | | D | S | | M | | | Xn | | |  |
| SUBA | B | W | L | 1 | 0 | 0 | 1 | An | | | S | 1 | 1 | M | | | Xn | | |  |
| SUBQ | B | W | L | 0 | 1 | 0 | 1 | Data | | | 1 | S | | M | | | Xn | | |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| Operation Size | Suffix | S | | S | S |  |  |  |
| Byte | .B | 0 | 0 | / | 0 | 1 |  |  |
| Word | .W | 0 | 1 | 0 | 1 | 1 |  |  |
| Long | .L | 1 | 0 | 1 | 1 | 0 |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Direction | dr |  |  |  |  |  |  |  |
| Shift Left | 1 |  |  |  |  |  |  |  |
| Shift Right | 0 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Direction | D |  |  |  |  |  |  |  |
| Dn \* <ea> -> Dn | 0 |  |  |  |  |  |  |  |
| <ea> \* Dn -> <ea> | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Instruction | ir |  |  |  |  |  |  |  |
| immediate shift count | 0 |  |  |  |  |  |  |  |
| register shift count | 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| COUNT Field | | | | | ir | COUNT | | Notes |
| contains shift count | | | | | 0 | 001-000 | | 000 = 8 |
| data register that contains the shift count | | | | | 1 | Dn | |  |
| **COUNT Field**  if ir = 0 - Contains the shift count. The values 1-7 represent counts of 1-7;  a value of zero represents a count of eight  if ir = 1 - Specifies the data register that contains the shift count | | | | | | | | |
| Addressing Mode | Format | M | | | Xn | | |  |
| Data Register | Dn | 0 | 0 | 0 | reg | | |  |
| Address Register | An | 0 | 0 | 1 | reg | | |  |
| Address | (An) | 0 | 1 | 0 | reg | | |  |
| Address with Postinc | (An)+ | 0 | 1 | 1 | reg | | |  |
| Address with Predec | -(An) | 1 | 0 | 0 | reg | | |  |
| Address with Displace | (d16, An) | 1 | 0 | 1 | reg | | |  |
| Address with Index | (d8, An, Xn) | 1 | 1 | 0 | reg | | |  |
| Program Counter with Displacement | (d16,PC) | 1 | 1 | 1 | 0 | 1 | 0 |  |
| Program Counter with Index | (d8, PC, Xn) | 1 | 1 | 1 | 0 | 1 | 1 |  |
| Absolute Short | (xxx).W | 1 | 1 | 1 | 0 | 0 | 0 |  |
| Absolute Long | (xxx).L | 1 | 1 | 1 | 0 | 0 | 1 |  |
| Immediate | #<data> | 1 | 1 | 1 | 1 | 0 | 0 |  |