

DSP FPGA comandos utiles

jmestanza

August 2020

1 Compilar un proyecto y simularlo

Suponiendo nuestro archivo se llama "ejemplo.asm", abrimos el cmd y vamos al directorio donde se encuentre el mismo. Alli corremos los siguientes comandos:

- asm ejemplo
- lnk ejemplo
- sim.bat
- load ejemplo.lod

Y con eso estamos dentro del simulador

2 Comandos útiles para la simulación

2.0.1 trace

Corre 1 linea de codigo. Si ponemos "trace 20", corre 20 lineas.

2.0.2 display

display: muestra un lugar en memoria o un bloque de memoria. Por ejemplo si quiero ver que hay en \$1234 de x

display x:\$1234

display x:\$1234..\$1238

2.0.3 evaluate

evaluate x:\$1234 da la informacion de lo que hay en esa direccion en binario, hexa y fraccionario.

3 Instruction set (1) motorola 56000 (detallado)

<https://drive.google.com/file/d/1d3Hb214fvF4eKKKghROFDbueV19gUNCG/view?usp=sharing>

4 Instruction set (2)

https://drive.google.com/file/d/1j9TeSJFOwoSgrzsoJUleeKuGnaWOy_3/view?usp=sharing

5 Instruction set (3) mirar Pagina 103

<https://drive.google.com/file/d/1gSdZPoGKH4xrGqmRRVyN8YryKMjuEGHW/view?usp=sharing>

6 Convergent Rounding

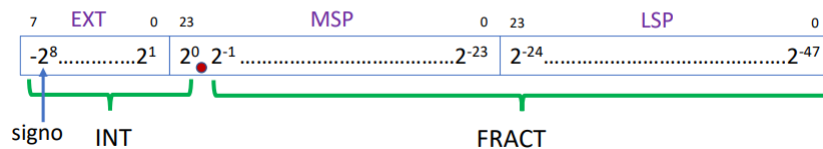
Si luego de hacer un mac (multiply and accumulate) vamos a guardar en una zona de memoria, siempre es bueno redondear (round - convergent rounding) antes de pasar a memoria (ver instruccion MACR). De esa forma evitamos directamente truncar y perder informacion valiosa.

Para ver como funciona convergent rounding visitar Instruction set(1) pagina 45.

7 Accumulator Format

La forma basica de un acumulador es la siguiente:

Accumulator Format



Cuidado con donde se encuentra el punto decimal.

Si queremos configurar bits de extension y otras cosas, la posta esta aca:

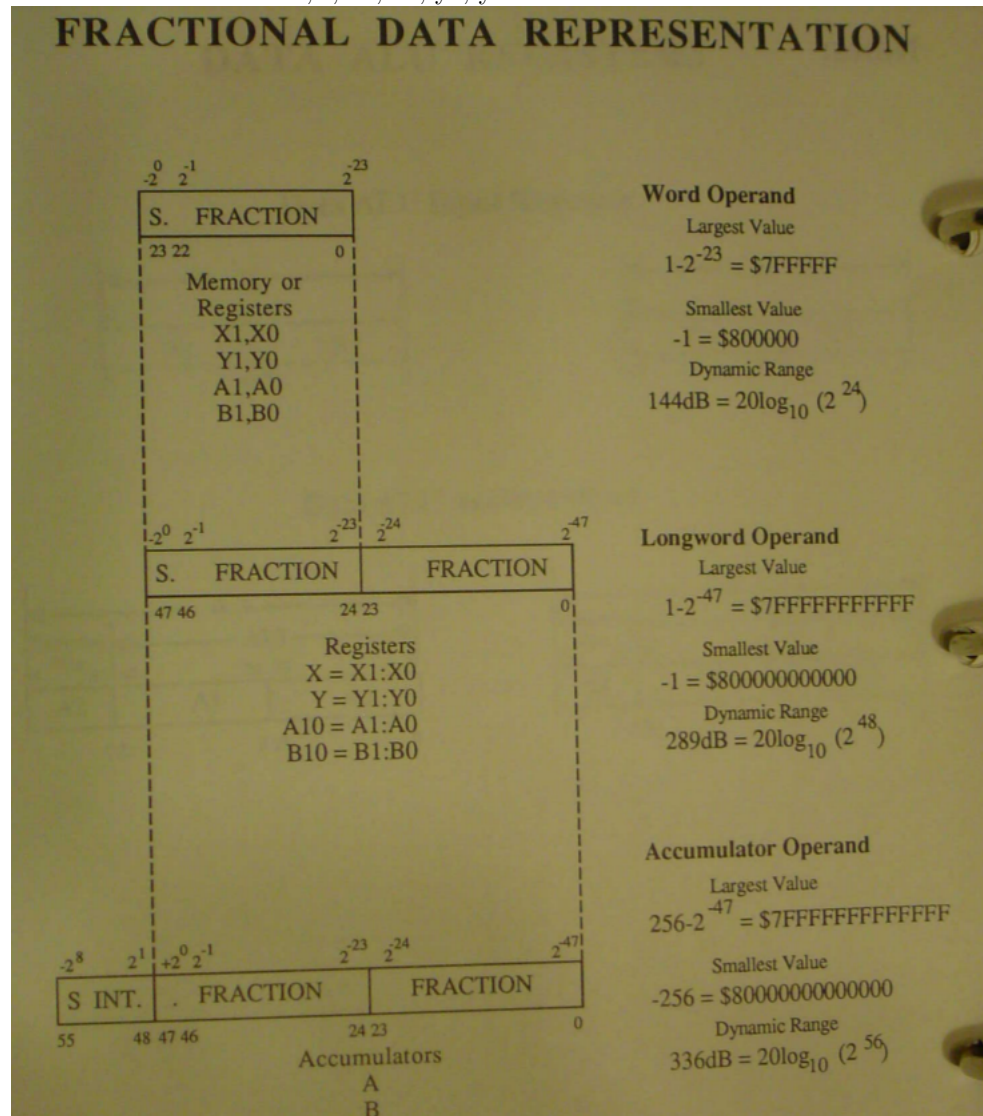
<https://drive.google.com/file/d/1VAkCF0ijhexpbEFBS3-dRCid960yflT7/view?usp=sharing>

8 Conditional transfers (Tcc)

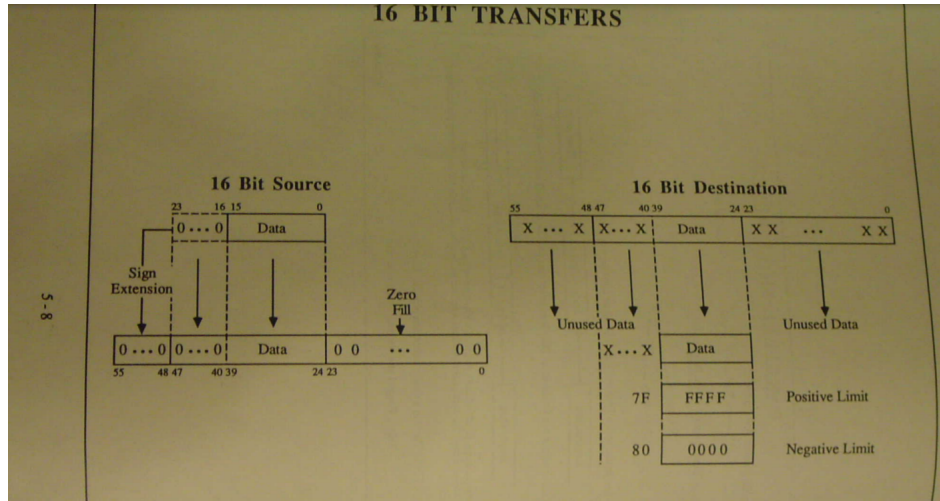
Pagina 106 de Instruction set (2)

9 Fractional data representation

Limites de acumulador A,B, x1, x0, y1, y0



10 16b transfers



11 Debugging: breakpoints

Cuando queremos debuggear, en el código ponemos `vvv` a la izquierda de todo de la línea que queremos poner breakpoint. Corremos `asm`, `lnk` y vamos a buscar al archivo generado que supongamos que se llama `ejemplo.lst`. Allí debemos buscar nuestro breakpoint y la dirección en el código donde se encuentra. Suponiendo que la dirección encontrada de `vvv` es `$E020`.

Abrimos el simulador y ponemos:

```
break P: $E020
```

```
go
```

luego de estos dos comandos, el simulador debe correr todas las líneas de código hasta la línea de código que pusimos `vvv`.