Answers to Self-Test Questions

Tutorial 8: Multiplication

……To Go Back to QUESTIONS Click Here

1.	The letter T in T Register stands for Temporary.
2.	The letter P in P register stands for Product.
3.	a) `LT #22h ; Immediate addressing not supported.` b) `LT AR4 ; Invalid operand.` c) `LT 58h ; Correct. Direct addressing.` d) `LT * ; Correct. Indirect addressing.`
4.	One way to make the value in the P register zero is to use the instruction `MPY #0`
5.	Sign-extension mode has no effect upon the instruction `MPY`. The value in the T register and the value of the operand are both treated as signed numbers.
6.	The most negative number we can use as intermediate data with the instruction multiply `MPY` is –4096 decimal (1000h).
7.	The most positive number we can use as immediate data with the instruction multiply `MPY` is +4095 (0FFFh).
8.	When using the instruction `MPY`, the value in the T register taken to be signed. It therefore lies in the range –32768 to +32767 decimal.
9.	On the TMS320F24x, no multiplication instruction uses the accumulator. Instead the instructions `MPY` and `MPYU` use the T Register and the P Register.
10.	a) `MPYU #8888h ; Immediate values not allowed.` b) `MPYU 22h ; Correct. Direct addressing.` c) `MPYU AR6 ; Auxiliary registers not allowed.` d) `MPYU * ; Correct. Indirect addressing.`
11.	The instruction `MPY` treats the contents of the T register and the operand to be signed values. This means FFFFh is taken to be –1 decimal. On the other hand, the instruction `MPYU` takes both the values to be multiplied to be unsigned e.g. FFFFh is taken to be the positive value 65535 decimal.
12.	To copy the value from the P register to the accumulator we use the instruction `PAC`.

13.	To save the 32-bit product of a multiplication at data memory addresses 200 and 201h we might write:
	`CLRC CNF ; Put block B0 into data memory.`
	`LDP #4h ; Access 200h to 27Fh.`
	`SPH 0h ; Save high word of P Register.`
	`SPL 1h ; Save low word of P Register.`

14.	When scaling an ADC measurement, we use the instruction `MPYU`. This is because the maximum value of an ADC conversion stored in ADCFIFO1 or ADCFIFO2 FFC0h, which represents a positive value. The instruction `MPY` would incorrectly treat FFC0h as a *negative* number.