

1. (1) Make memory access instructions two times faster.
 (2) $20 * \frac{1}{4} + 50 * \frac{1}{2} + 30 = 60$
 $100 / 60 = 5 / 3 (1.66\dots)$
2. (a) Push x1 (ra) 是因為在該function中有可能會用jal類的指令call其他function，如此以來ra會被覆蓋，所以要Push進去；Push x10 (a0) 是因為該function可能會要傳argument給其它function，導致a0被覆蓋，所以也要把a0 push進去
 (b) `bge x5,x0,+16(8) => 0 000000 00000 00101 101 1000 0 1100011`
 (c) beq與jal的主要差異在於兩者之間的immediate的長度並不相同。而jal因為immediate的長度比beq長，所以jal所能跳的相對距離比beq還要遠。
3. `x1 = 0x000000005F82C95F`
`x2 = 0xFFFFFFFFFFFF000FF`
`x3 = 0x35ED293045BCE0F4`
 (a) `x1 = 1`
 (b) `x1 = 0xFFFF FFFF FFFE 001F`
 (c) `x1 = 0x07FF FFFF FFFF 8007 = 17592186044415`
4. (a) `x5 = 30`
 (b) 61 instructions
5. (a) 要做到slt(set on less than)，也就是去測試A是否小於B，只要使A減掉B (Ainvert = 0, Bnegate = 1, Op = 11)後去檢查結果是否為負數(A < B -> A - B < 0)，也就是去看sign bit是否為1(Set --> Result0)就可以知道。
 (b) 因為在overflow的時候，A - B的sign bit會有錯誤，而這個sign bit被拿來當作slt的结果自然也會發生錯誤
6. (a) 如下圖有色部分 (b) 因為最後面結果出來時餘數會被多shift left一次，如果不shift right的話，餘數會錯掉 (c) 14 cycles

step	Remainder	Divisor	Cycles
0	0000 0111 (1 cycle)	0011	1
1.1	0000 1110 (1 cycle)	0011	1
1.2	1101 1110 (1 cycle)	0011	1
1.3b	0001 1100 (2 cycle)	0011	2
2.2	1110 1100 (1 cycle)	0011	1
2.3b	0011 1000 (2 cycle)	0011	2
3.2	0000 1000 (1 cycle)	0011	1
3.3a	0001 0001 (1 cycle)	0011	1
4.2	1110 0001 (1 cycle)	0011	1
4.3b	0010 0010 (2 cycle)	0011	2
Done	0001 0010 (1 cycle)	0011	1

7. (a) $35.75 = 0100\ 0010\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000$
- (b) $0x738E9BDO = 0\ 11100111\ 00011101001101111010000$
 $0x058A\ B913 = 0\ 00001011\ 00010101011100100010011$
 S Exponent Significand

$$0x738E9BDO = (1.00011101001101111010000) * 2^{(231-127)}$$

$$0x058A\ B913 = (1.00010101011100100010011) * 2^{(11-127)}$$

所以0x738E9BDO是比較大的

(c) $0x738E9BDO = 1,938,725,840$

$0x058A\ B913 = 92,977,427$

所以0x738E9BDO是比較大的

- (d) 為了避免最小的normalized number與最大的denormalized number的差異過大