```
struct{
   int     a;    //0x1112_1314                    word
   int     pad;  //
   double  b;    //0x2122_2324_2526_2728          doubleword
   char*   c;    //0x3132_3334                    word
   char    d[7]; //'A'.'B','C','D','E','F','G'    byte array
   short   e;    //0x5152                         halfword
   int     f;    //0x6161_6364                    word
} s;
```

Big-endian address mapping

Byte Address								
	11	**12**	**13**	**14**				
00	00	01	02	03	04	05	06	07
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**
08	08	09	0A	0B	0C	0D	0E	0F
	31	**32**	**33**	**34**	**'A'**	**'B'**	**'C'**	**'D'**
10	10	11	12	13	14	15	16	17
	'E'	**'F'**	**'G'**		**51**	**52**		
18	18	19	1A	1B	1C	1D	1E	1F
	61	**62**	**63**	**64**				
20	20	21	22	23				

Little-endian address mapping

								Byte Address
				11	**12**	**13**	**14**	
07	06	05	04	03	02	01	00	00
21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	
0F	0E	0D	0C	0B	0A	09	08	08
'D'	**'C'**	**'B'**	**'A'**	**31**	**32**	**33**	**34**	
17	16	15	14	13	12	11	10	10
		51	**52**		**'G'**	**'F'**	**'E'**	
1F	1E	1D	1C	1B	1A	19	18	18
				61	**62**	**63**	**64**	
				23	22	21	20	20

Figure 9.18 Example C Data Structure and its Endian Maps [IBM94]