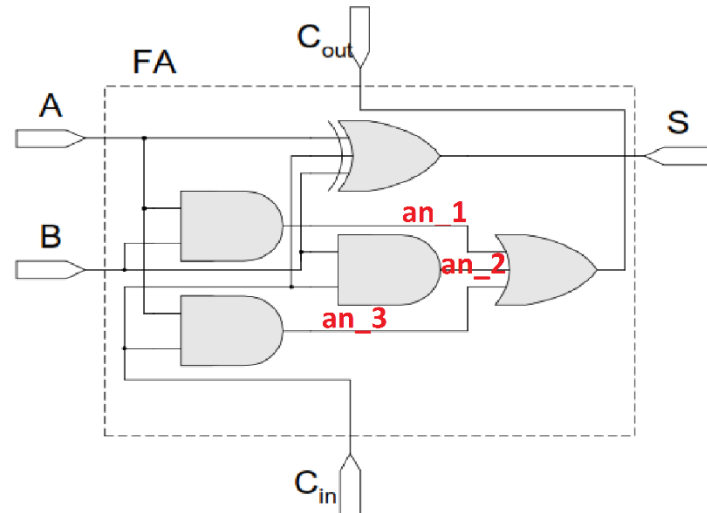


## Ερωτήματα:

### 1. α) structural περιγραφή του full adder

Σχήμα Full Adder με τα αντίστοιχα ονόματα καλωδίων:



Κώδικας Verilog του αρχείου FAStruct.v:

```
module FAStruct(input a,b,cin,output sum,cout);
    wire an_1,an_2,an_3;
    and(an_1,a,b);
    and(an_3,a,cin);
    and(an_2,b,cin);
    or(cout,an_1,an_2,an_3);
    xor(sum,a,b,cin);
endmodule
```

### β) behavioral περιγραφή του full adder

Κώδικας Verilog του αρχείου FABehav.v:

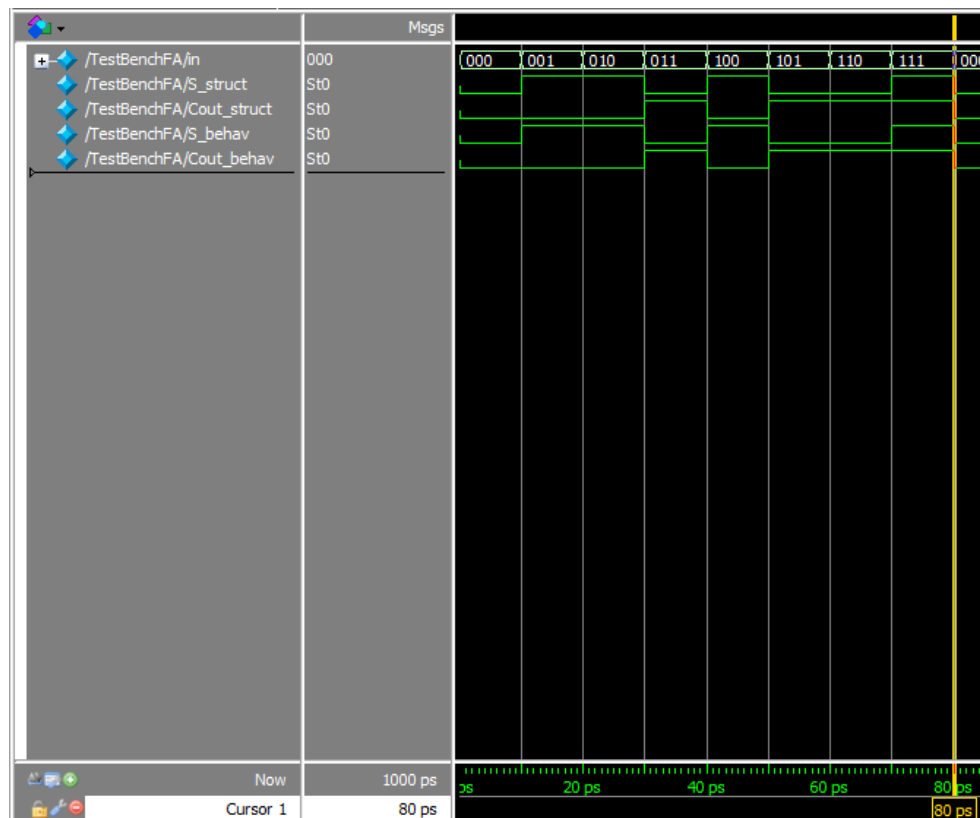
```
module FABehav(input a,b,cin,output sum,cout);
    assign {cout,sum} = a + b + cin;
endmodule
```

2. Στην συνέχεια εξομοιώνουμε και τις δύο περιγραφές με το παρακάτω test bench.

**Κώδικας Verilog του αρχείου TestBenchFA.v:**

```
module TestBenchFA;
    reg [2:0] in;
    wire S_struct, Cout_struct;
    wire S_behav, Cout_behav;
    FAStruct fa0(in[2], in[1], in[0], S_struct, Cout_struct);
    FABehav fa1(in[2], in[1], in[0], S_behav, Cout_behav);
    initial
        in = 3'b000;
    always
        #10 in = in + 1;
endmodule
```

Παρακάτω φαίνονται τα σήματα που παίρνουμε για τις εξόδους S, Cout για όλες τις τιμές τριών bit του in και για τις δύο περιγραφές. Τα S\_struct, Cout\_struct αφορούν την structural περιγραφή, ενώ τα S\_behav Cout\_behav αφορούν την behavioral περιγραφή.



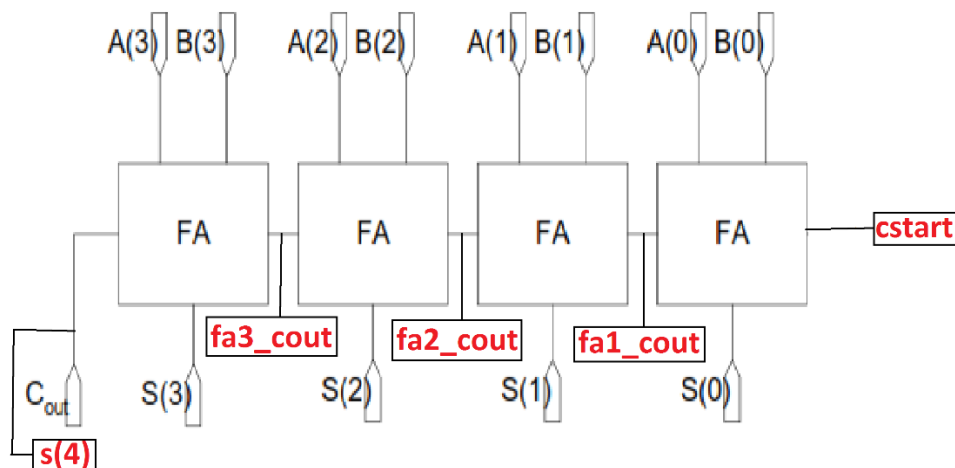
Παρατηρούμε πως και στις δύο περιγραφές παίρνουμε ακριβώς τα ίδια σήματα το οποίο είναι αναμενόμενο αφού περιγράφουμε ακριβώς το ίδιο κύκλωμα. Τέλος συγκρίνουμε τα αποτελέσματα με τον πίνακα αλήθειας του full adder που φαίνεται παρακάτω.

### Πίνακας αλήθειας του Full Adder

A	B	C <sub>in</sub>	Sum	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3. Σύμφωνα με το παρακάτω σχήμα του 4-bit ripple carry adder που μας δίνεται φτιάχνουμε το αντίστοιχο αρχείο Verilog.

### 4-bit Ripple Carry Adder



```
module FourBitRippleCarryAdder(a,b,cstart,s);
```

endmodule








που φαίνεται παρακάτω.

**TestBenchFourBitRippleCarryAdder.v:**








endmodule

Παρακάτω βλέπουμε ενδεικτικά 3 screenshots από την εξομίωση.

	Msgs																		
+ /TestBenchFourBitRippleCarryAdder/a	6	2																	
+ /TestBenchFourBitRippleCarryAdder/b	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
+ /TestBenchFourBitRippleCarryAdder/S	21	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		

			Msgs																					
		/TestBenchFourBitRippleCarryAdder/a	6	5	6																			
		/TestBenchFourBitRippleCarryAdder/b	15	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
		/TestBenchFourBitRippleCarryAdder/S	21	20	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21				

			Msgs																															
		/TestBenchFourBitRippleCarryAdder/a	6	14																15														
		/TestBenchFourBitRippleCarryAdder/b	15	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15						
		/TestBenchFourBitRippleCarryAdder/S	21	21	22	23	24	25	26	27	28	29	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						

**4.** Τέλος χρησιμοποιώντας δύο 4-bit αθροιστές που φτιάξαμε στο 3<sup>ο</sup> ερώτημα δημιουργούμε έναν 8-bit αθροιστή.

### Κώδικας Verilog του αρχείου EightBitRippleCarryAdder.v:

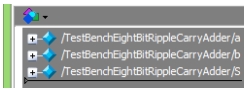
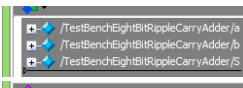

```
module EightBitRippleCarryAdder(a,b,cstart,s);
    output [8:0] s;
    input [7:0] a,b;
    input cstart;
    wire [9:0] c;
    FourBitRippleCarryAdder fbrca1(a[3:0],b[3:0],cstart,c[4:0]);
    FourBitRippleCarryAdder fbrca2(a[7:4],b[7:4],c[4],c[9:5]);
    assign s[3:0]=c[3:0];
    assign s[8:4]=c[9:5];
endmodule
```

### Κώδικας Verilog του αρχείου

### TestBenchEightBitRippleCarryAdder.v:

```
module TestBenchEightBitRippleCarryAdder;
    reg [7:0] a,b;
    wire [8:0] S;
    EightBitRippleCarryAdder ebrca(a,b,1'b0,S);
    initial begin
        a = 8'b00000000;
        b = 8'b00001111;
    end
    always
        #20 a <= a + 1;
    always
        #10 b <= b + 1;
endmodule
```

Παρακάτω φαίνονται 3 ενδεικτικά screenshots από την εξομοίωση του 8-bit αθροιστή:

		Mags																																																																																													
	<pre>//TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s</pre>	124 8 132																																																																																													
<table><tr><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td></tr><tr><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td><td>81</td><td>82</td><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td><td>90</td><td>91</td><td>92</td><td>93</td><td>94</td><td>95</td><td>96</td><td>97</td><td>98</td><td>99</td></tr><tr><td>87</td><td>88</td><td>90</td><td>91</td><td>93</td><td>94</td><td>96</td><td>97</td><td>99</td><td>100</td><td>102</td><td>103</td><td>105</td><td>106</td><td>108</td><td>109</td><td>111</td><td>112</td><td>114</td><td>115</td><td>117</td><td>118</td><td>120</td><td>121</td><td>123</td><td>124</td><td>126</td><td>127</td><td>129</td><td>130</td><td>132</td><td>133</td><td>135</td><td>136</td><td>138</td><td>139</td><td>141</td></tr></table>			24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	87	88	90	91	93	94	96	97	99	100	102	103	105	106	108	109	111	112	114	115	117	118	120	121	123	124	126	127	129	130	132	133	135	136	138	139	141
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42																																																																													
63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99																																																											
87	88	90	91	93	94	96	97	99	100	102	103	105	106	108	109	111	112	114	115	117	118	120	121	123	124	126	127	129	130	132	133	135	136	138	139	141																																																											
		Mags																																																																																													
	<pre>//TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s</pre>	124 8 132																																																																																													
<table><tr><td>66</td><td>67</td><td>68</td><td>69</td><td>70</td><td>71</td><td>72</td><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td><td>81</td><td>82</td><td>83</td><td>84</td></tr><tr><td>148</td><td>149</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td><td>157</td><td>158</td><td>159</td><td>160</td><td>161</td><td>162</td><td>163</td><td>164</td><td>165</td><td>166</td><td>167</td><td>168</td><td>169</td><td>170</td><td>171</td><td>172</td><td>173</td><td>174</td><td>175</td><td>176</td><td>177</td><td>178</td><td>179</td><td>180</td><td>181</td><td>182</td><td>183</td><td>184</td></tr><tr><td>214</td><td>216</td><td>217</td><td>219</td><td>220</td><td>222</td><td>223</td><td>225</td><td>226</td><td>228</td><td>229</td><td>231</td><td>232</td><td>234</td><td>235</td><td>237</td><td>238</td><td>240</td><td>241</td><td>243</td><td>244</td><td>246</td><td>247</td><td>249</td><td>250</td><td>252</td><td>253</td><td>255</td><td>256</td><td>258</td><td>259</td><td>261</td><td>262</td><td>264</td><td>265</td><td>267</td><td>268</td></tr></table>			66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	214	216	217	219	220	222	223	225	226	228	229	231	232	234	235	237	238	240	241	243	244	246	247	249	250	252	253	255	256	258	259	261	262	264	265	267	268
66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84																																																																													
148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184																																																											
214	216	217	219	220	222	223	225	226	228	229	231	232	234	235	237	238	240	241	243	244	246	247	249	250	252	253	255	256	258	259	261	262	264	265	267	268																																																											
		Mags																																																																																													
	<pre>//TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s //TestBenchEightBitRippleCarryAdder.s</pre>	243 246 489																																																																																													
<table><tr><td>226</td><td>227</td><td>228</td><td>229</td><td>230</td><td>231</td><td>232</td><td>233</td><td>234</td><td>235</td><td>236</td><td>237</td><td>238</td><td>239</td><td>240</td><td>241</td><td>242</td><td>243</td></tr><tr><td>212</td><td>213</td><td>214</td><td>215</td><td>216</td><td>217</td><td>218</td><td>219</td><td>220</td><td>221</td><td>222</td><td>223</td><td>224</td><td>225</td><td>226</td><td>227</td><td>228</td><td>229</td><td>230</td><td>231</td><td>232</td><td>233</td><td>234</td><td>235</td><td>236</td><td>237</td><td>238</td><td>239</td><td>240</td><td>241</td><td>242</td><td>243</td><td>244</td><td>245</td><td>246</td></tr><tr><td>438</td><td>440</td><td>441</td><td>443</td><td>444</td><td>446</td><td>447</td><td>449</td><td>450</td><td>452</td><td>453</td><td>455</td><td>456</td><td>458</td><td>459</td><td>461</td><td>462</td><td>464</td><td>465</td><td>467</td><td>468</td><td>470</td><td>471</td><td>473</td><td>474</td><td>476</td><td>477</td><td>479</td><td>480</td><td>482</td><td>483</td><td>485</td><td>486</td><td>488</td><td>489</td></tr></table>			226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	438	440	441	443	444	446	447	449	450	452	453	455	456	458	459	461	462	464	465	467	468	470	471	473	474	476	477	479	480	482	483	485	486	488	489					
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243																																																																														
212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246																																																													
438	440	441	443	444	446	447	449	450	452	453	455	456	458	459	461	462	464	465	467	468	470	471	473	474	476	477	479	480	482	483	485	486	488	489																																																													