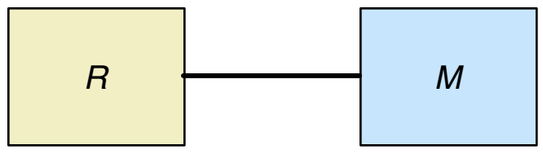
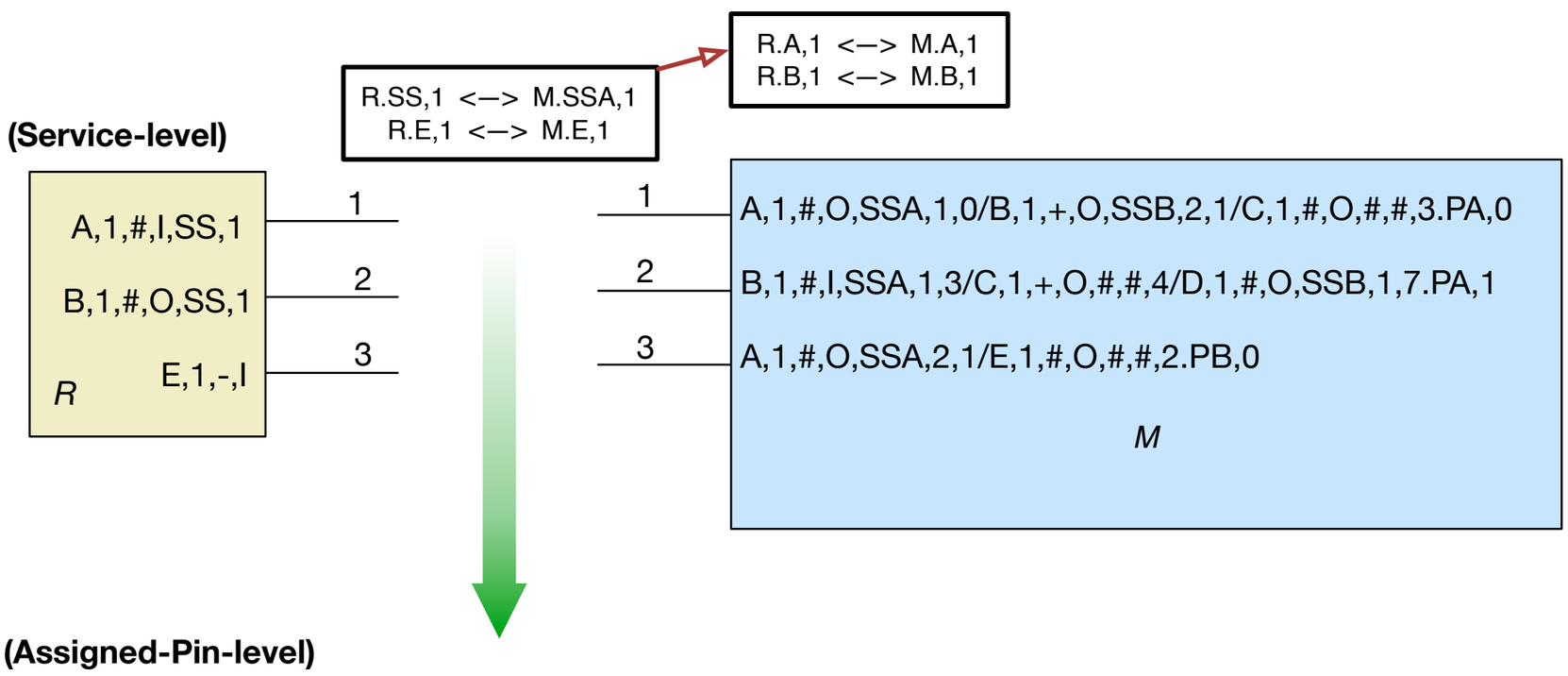
L_0 (User-level)



L_1 (Service-level)



L_2 (Assigned-Pin-level)

R.1.A,1,#,I,SS,1 <—> M.1.A,1,#,O,SSA,1,0.PA,0
R.2.B,1,#,O,SS,1 <—> M.2.B,1,#,I,SSA,1,3.PA,1
R.3.E,1,-,I <—> M.3.E,1,#,O,#,#,2.PB,0

Labeling Convention

C **.** PINOUT **.** Service **,** Service # **,** 1-to-many-symbol (+ -) **,** input-output-symbol (I and/or O) **,** SuperService **,** SuperService # **,** AF **/** … **.** PORT , PININ

C: Component name
PINOUT: Component Pin number
Service: Component service name
Service #: Number of a particular service on an individual component
1-to-many-symbol: + for a 1-to-many port, - for a single port which dominates a 1-to-many port if connected to one
input-output-symbol: I for an input port, O for an output port, IO for both input and output port
SuperService: Name for an encapsulation of multiple Services on multiple ports
SuperService #: Number of a particular SuperService on an individual component
AF: Alternate Function number used on certain components to specify which Service is assigned for the pin
PORT: Internal component port name
PININ: Internal component pin number

Use ‘#’ character for null values; null values occurring at the end of a comma-delimited sequence may be dropped
For bus connections: the shared connection name becomes **SuperService,Service,+/-,<unique id number>**

L_3 (Output-level)

.h example syntax (#defines)

```
# R_A_1_Port   PA   // port
# R_A_1_Pin    0    // PININ
# R_A_1_AF     0    // AF
```

Internal ROSLab data structure representations of electronics pin-service & port-AF mappings for a single component (M in this example)

PINOUT / Service	1	2	3
A	A,1,#,O,SSA,1,0		A,1,#,O,SSA,2,1
B	B,1,+,O,SSB,2,1	B,1,#,I,SSA,1,3	
C	C,1,#,O,#,#,3	C,1,+,O,#,#,4	
D		D,1,#,O,SSB,1,7	
E			E,1,#,I,#,#,2

AF / Port	AF0	AF1	AF2	AF3	AF4		AF7
PA,0	A,1	B,1		C,1			
PA,1				B,1	C,1		D,1
PB,0		A,1	E,1				