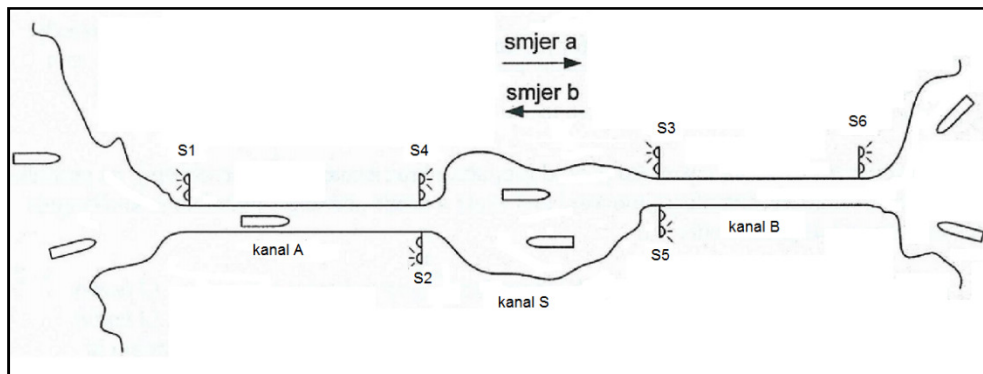


% konflikt B  
AKO  $(m(I_2) > 0) \ \& \ (m(M_1) > 1) \ \& \ (m(B) = 1)$  ONDA  
     $u_3 = 1$   
     $u_4 = 0$   
INAČE  
     $u_3 = 1$   
     $u_4 = 1$   
KRAJ

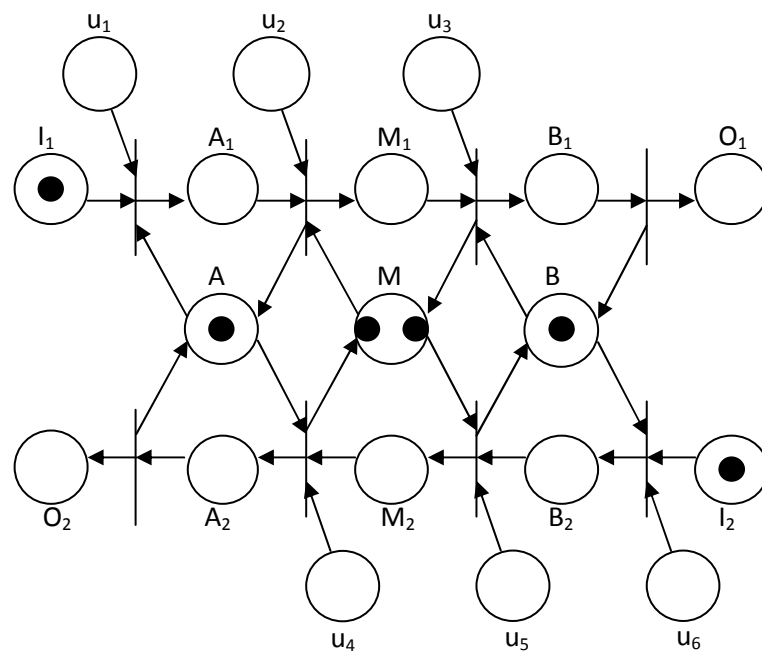
e) (algoritam 3) Odrediti upravljački algoritam kojim se sprečavaju zaglavljenja i razrješavaju konflikti na temelju određenih kritičnih podsustava.

Rješenje:

Iako i algoritmi određeni u d) zadatku sprečavaju zaglavljenje (statičke) Petrijeve mreže zadanog sustava (objasniti zašto!), ovdje će biti prikazan korak-po-korak način na koji se općenito može odrediti upravljanje na temelju kritičnih podsustava. Da bi sadržaj svih kritičnih sustava bio potpuno upravljiv iz upravljačkog algoritma, potrebno je dodati semafore na prijelaze iz kanala A i B u središnji kanal. Tako dobiveni sustav prikazan je na Slici 1, a pripadna Petrijeva mreža na Slici 2:



Slika 1. Sustav s dodanim semaforima



Slika 2. Petrijeva mreža sustava

Postoji više ispravnih načina da se definira upravljački algoritam. Cilj je uvijek da svaka potencijalna kritična situacija (potencijalno zaglavljenje ili konflikt) bude ispravno regulirana.

$u_1=u_2=u_3=u_4=u_5=u_6=1$

AKO  $(m(A_1)=0) \ \& \ (m(M_2)=2)$  ONDA  
%zaglavljenje kružno čekanje C1  
%konflikt A  
 $u_1=0$

INAČE AKO  $(m(A_1)=1) \ \& \ (m(M_1)=1) \ \& \ (m(M_2)=1)$  ONDA  
%zaglavljenje kružno čekanje C3  
%konflikt B  
 $u_6=0$

INAČE AKO  $(m(A_1)=1) \ \& \ (m(M_2)=1)$  ONDA  
%zaglavljenje kružno čekanje C1  
 $u_5=0$

INAČE AKO  $(m(I_1)>0) \ \& \ (m(M_2)=1)$  ONDA  
%konflikt A, ako prioritet ima punjenje sustava  
 $u_4=0$

KRAJ

AKO  $(m(B_2)=0) \ \& \ (m(M_1)=2)$  ONDA  
%zaglavljenje kružno čekanje C2  
%konflikt B  
 $u_6=0$

INAČE AKO  $(m(B_2)=1) \ \& \ (m(M_1)=1) \ \& \ (m(M_2)=1)$  ONDA  
%zaglavljenje kružno čekanje C3  
%konflikt A  
 $u_1=0$

INAČE AKO  $(m(B_2)=1) \ \& \ (m(M_1)=1)$  ONDA  
%zaglavljenje kružno čekanje C2  
 $u_2=0$

INAČE AKO  $(m(M_1)=1) \ \& \ (m(I_2)>0)$  ONDA  
%konflikt B, ako prioritet ima punjenje sustava  
 $u_3=0$

KRAJ

Svi upravljački algoritmi iz d) i e) izbjegavaju zaglavljenja i razrješavaju konflikte. Razlikuju se po broju oznaka koji istovremeno biti u sustavu. U tom pogledu algoritam 1 je najrestriktivniji, a algoritam 3 najmanje restriktivan.