

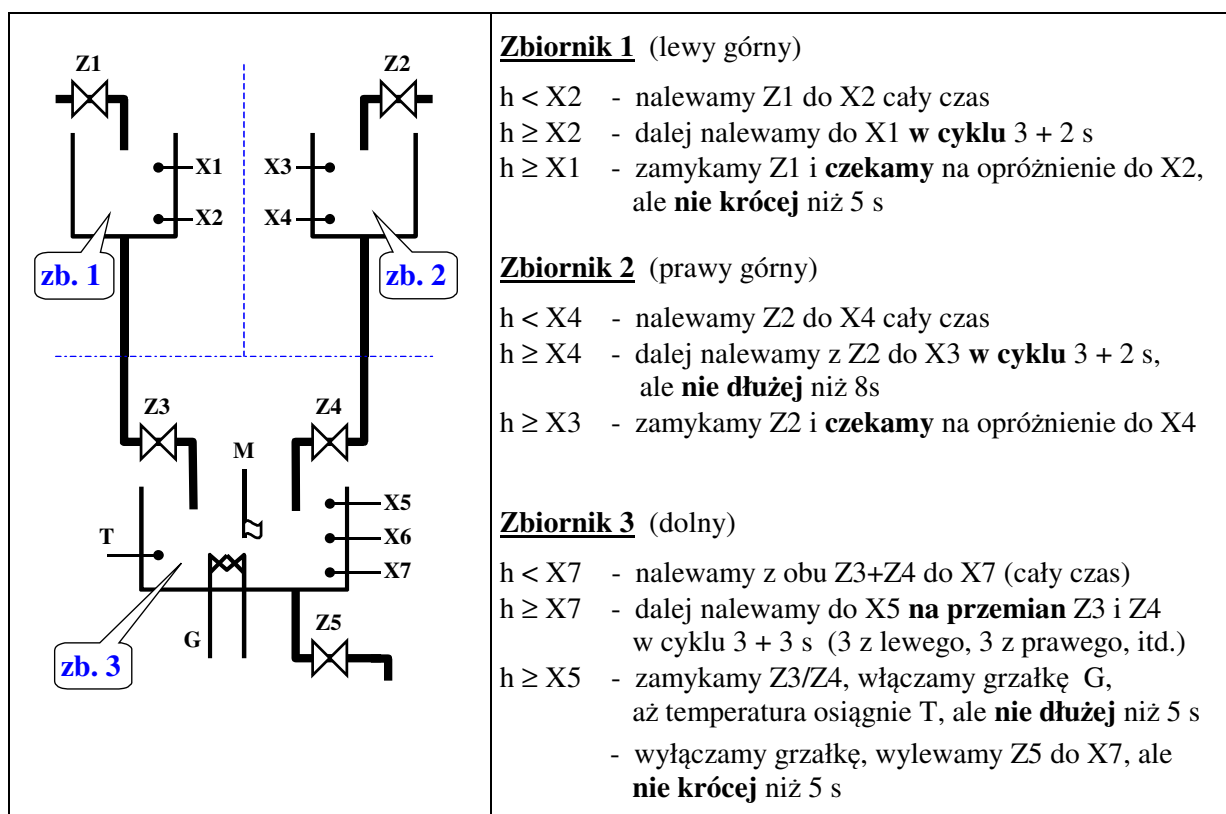
Materiały do przedmiotu "Sterowniki mikroprocesorowe"

Opracował: Zbigniew Świder

Układy sekwencyjno-czasowe, cz. III (kaskada zbiorników)

5. Kaskada zbiorników

Zaprojektuj układ sterowania trzema zbiornikami według podanego algorytmu:

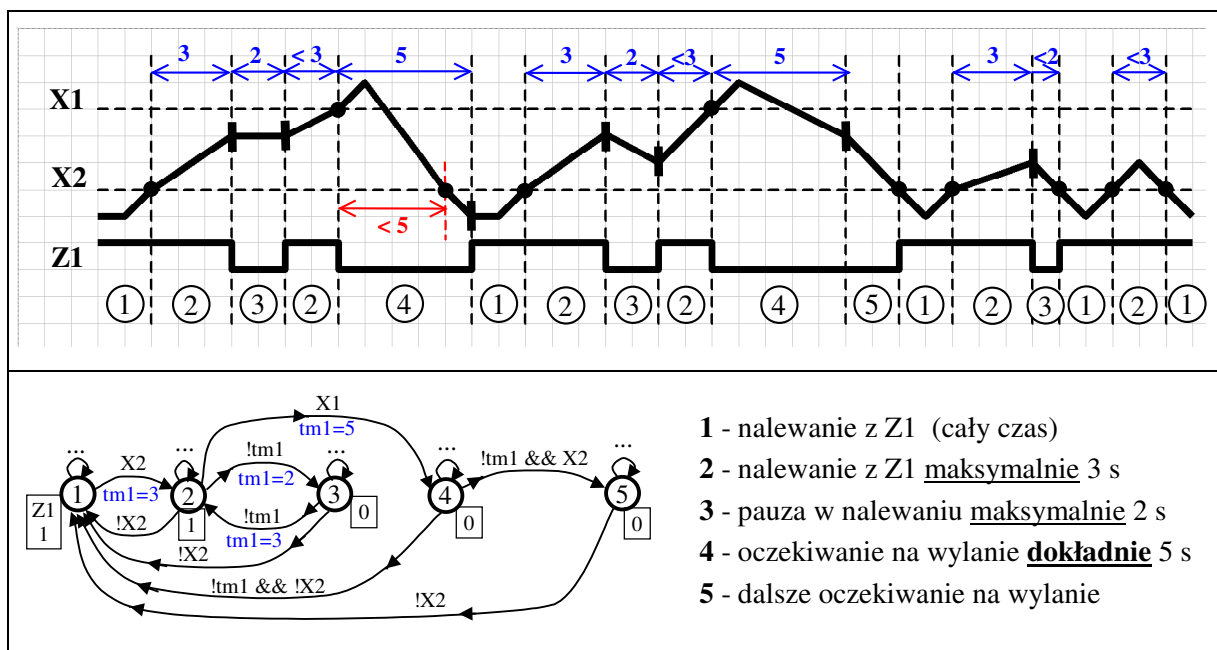


Rozwiązanie.

Dla tego typu kaskady zbiorników układ sterowania projektujemy **osobno** (niezależnie) dla każdego zbiornika. Należy więc najpierw narysować przebiegi czasowe, opis stanów i graf dla pierwszego zbiornika, potem przebiegi czasowe, opis stanów i graf dla drugiego zbiornika, a następnie przebiegi czasowe, opis stanów i graf dla trzeciego zbiornika. Na koniec należy napisać **jeden** program sterujący wszystkimi trzema zbiornikami (zawierający trzy instrukcje *switch* - po jednej dla każdego zbiornika).

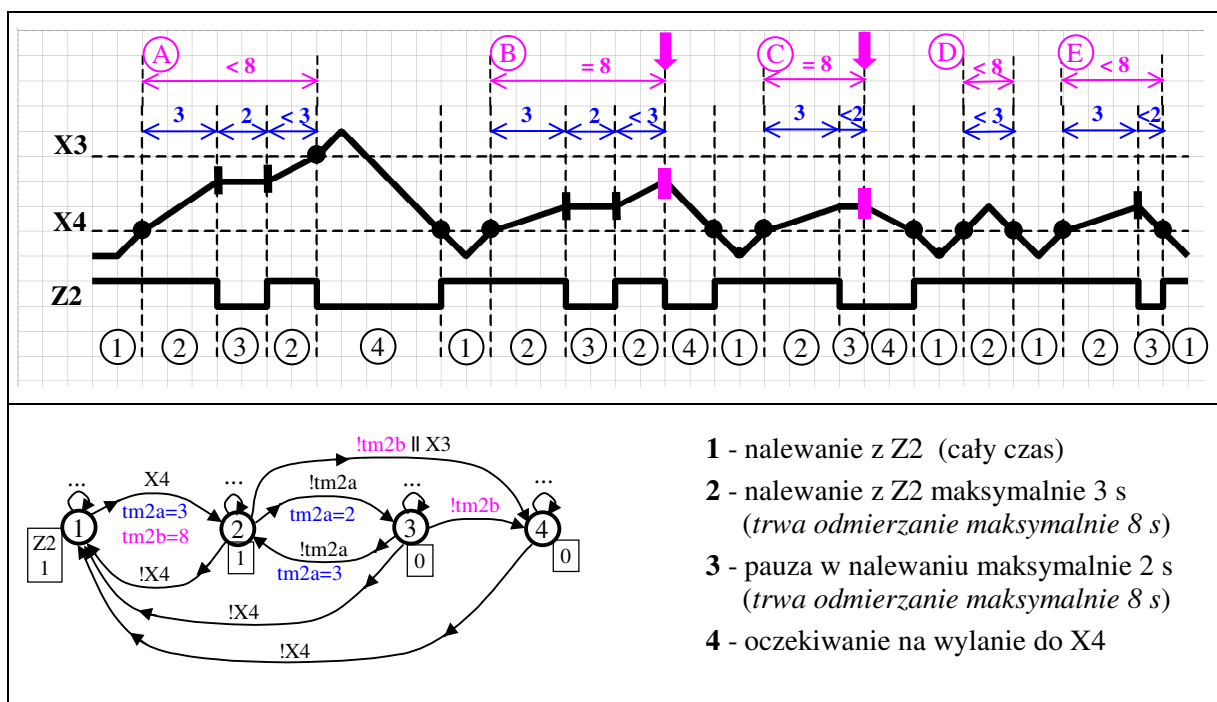
Zwróćmy również uwagę, że zawory Z3 i Z4 "przynależą" do **dolnego zbiornika** (są sterowane przez trzeci graf). Należy także uwzględnić sytuację, kiedy to w górnych zbiornikach woda będzie się **szybciej wylewać niż nalewać** (jeśli akuratnie trzeci graf otworzy zawory Z3 lub Z4).

Zbiornik 1 (lewy górny)



Pamiętamy o uwzględnieniu sytuacji, że w górnym zbiorniku woda może się szybciej wylewać niż nalać (na końcu przebiegów), i to zarówno podczas 3-sekundowego nalewania z Z1, jak i podczas 2-sekundowej pauzy. Zawór Z3 przynależy do dolnego zbiornika, dlatego go tutaj nie rysujemy (jest on traktowany jako zakłócenie). Pamiętamy również o zastosowaniu iloczynu z timerem w warunkach przejścia dla stanu 4 (jest "dokładnie").

Zbiornik 2 (prawy górny)



Po analizie zadania, na przebiegu można wyróżnić pięć możliwości nalewania w cyklu 3+2:

- (A) - **zdażyło się nalać** w mniej niż 8 sekund (ostatnie nalewanie "mniej niż 3s")
- (B) - **nie zdażyło się nalać** w 8 sekund, czas się skończył podczas **nalewania** z Z2 (= 8s)
- (C) - **nie zdażyło się nalać** w 8 sekund, czas się skończył podczas **pauzy** w nalewaniu (= 8s)

- (D) - **szybciej się wylewa niż nalewa**, wylało się w trakcie nalewania z Z2 (< 8s)
 (E) - **szybciej się wylewa niż nalewa**, wylało się w trakcie paury w nalewaniu (< 8s).

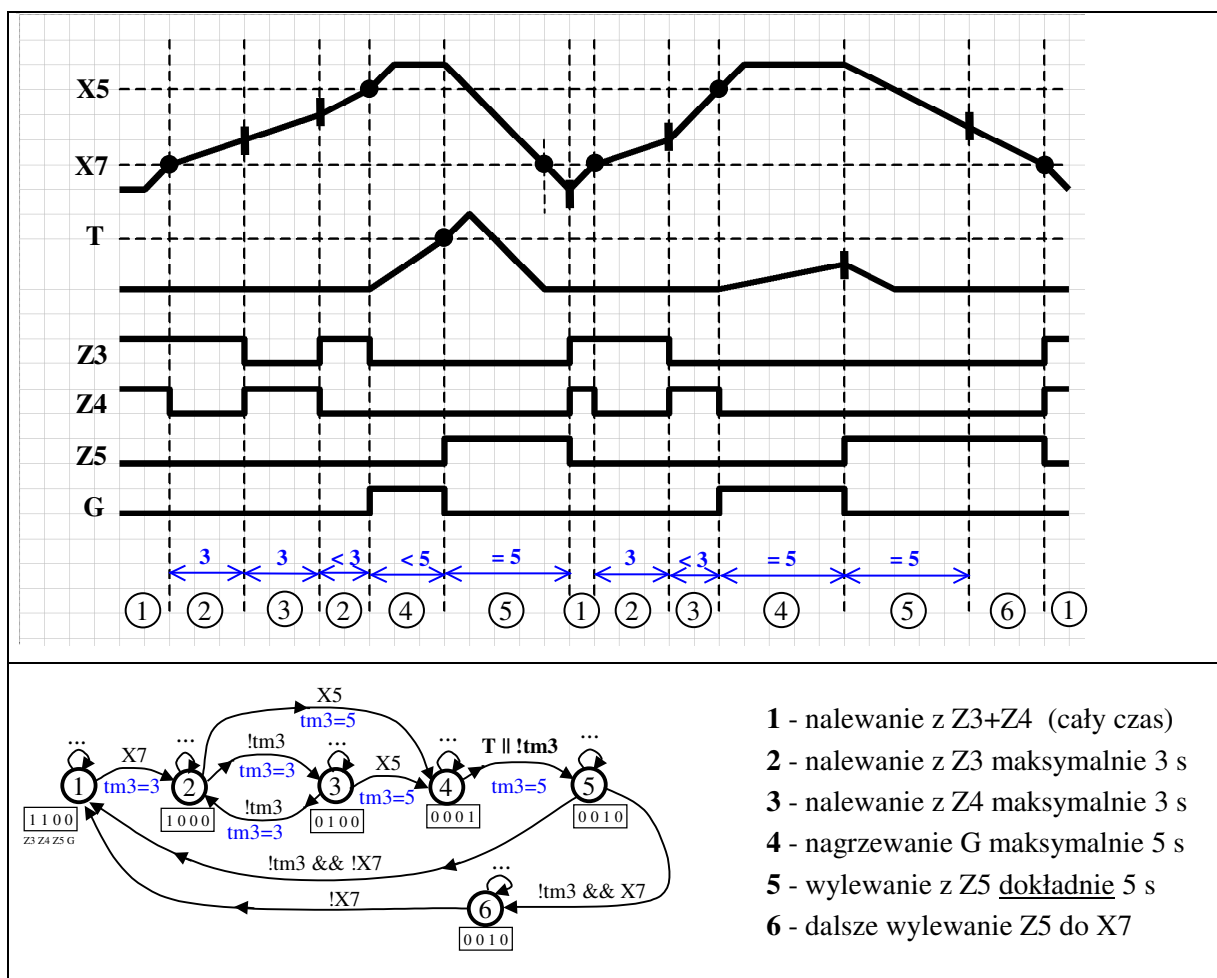
Zwróćmy szczególną uwagę na przypadek C. Jest to sytuacja, gdy czas 8s ($tm2b$) skończył się podczas trwania 2s paury w nalewaniu ($tm2a$). Pomimo, że z wyliczeń czasów wynika, że 8s teoretycznie zawsze się skończy podczas nalewania w cyklu 3+2, to **taki przypadek też musimy rozpatrzyć**.

Dotychczas w zadaniach czasy są podane "na sztywno" (np. "8 sekund"), ale w kolejnych będą mogły być zmieniane w trakcie pracy sterownika (np. z panelu operatorskiego czy łączem komunikacyjnym), dlatego też **MUSIMY** rozpatrzyć wszystkie możliwości i **NIE WOLNO** wyliczać na podstawie czasów, czy koniec czasu wypadnie podczas nalewania czy też paury - zawsze należy rysować **OBA** przypadki. Wtedy oczywiście nie będzie zachowana zasada "1 kratka = 1 sekunda", bo inaczej nie da się tego narysować (patrz przebiegi dla przypadku C - rysujemy wtedy na 4 kratki i piszemy "8 s").

Przy przejściu ze stanu 1 do 2 uruchamiamy jednocześnie dwa timery: pierwszy $tm2a$ odmierzają cykl 3+2, natomiast drugi $tm2b$ odmierzają czas 8 sekund (patrz opis stanów). Tutaj oba timery jednocześnie odmierzają czas (są dekrementowane co cykl).

Zwróćmy jeszcze uwagę na warunek przejścia ze stanu 2 do 4 ($!tm2b \parallel X3$). Przejście to nastąpi, gdy woda naleje się do pełna (A) **lub** skończy się czas (B,C). Ze stanu 2 możemy również przejść do 1, kiedy to po otwarciu Z4 przez trzeci graf, ze zbiornika drugiego wyleje się woda poniżej poziomu X4.

Zbiornik 3 (dolny)



Tu również zwróćmy uwagę na warunek przejścia ze stanu 4 do 5 ($T \parallel !tm3$) - przejście nastąpi w dwóch sytuacjach, czyli gdy osiągnie temperaturę T **lub** też skończy się czas.

Program w języku C:

```
char X1, X2, X3, X4, X5, X7, T, Z1, Z2, Z3, Z4, Z5, G;
char stan1, tm1, stan2=1, tm2a, tm2b, stan3=1, tm3;

// -----

X1=aK1; X2=aK2; X3=aK3; X4=aK4; X5=aK5; X7=aK6; T=aK7;

switch(stan1) { // ----- zbiornik 1
    case 1: Z1=1; if(X2){tm1=30; stan1=2;}
             break;
    case 2: Z1=1; if(X1){tm1=50; stan1=4;}
             else if(!tm1){tm1=20; stan1=3;}
             else if(!X2) stan1=1;
             break;
    case 3: Z1=0; if(!tm1){tm1=30; stan1=2;}
             else if(!X2) stan1=1;
             break;
    case 4: Z1=0; if(!tm1&&X2) stan1=5;
             else if(!tm1&&!X2) stan1=1;
             break;
    case 5: Z1=0; if(!X2) stan1=1;
             break;
}
if (tm1) --tm1;

switch(stan2) { // ----- zbiornik 2
    case 1: Z2=1; if(X4){tm2a=30; tm2b=80; stan2=2;}
             break;
    case 2: Z2=1; if(tm2b||X3) stan2=4;
             else if(!tm2a){tm2a=20; stan2=3;}
             else if(!X4) stan2=1;
             break;
    case 3: Z2=0; if(!tm2a){tm1=30; stan2=2;}
             else if(!tm2b) stan2=4;
             else if(!X4) stan2=1;
             break;
    case 4: Z2=0; if(!X4) stan2=1;
             break;
}
if (tm2a) --tm2a;
if (tm2b) --tm2b;

switch(stan3) { // ----- zbiornik 3
    case 1: Z3=1; Z4=1; Z5=0; G=0;
             if(X7){tm3=30; stan3=2;}
             break;
    case 2: Z3=1; Z4=0; Z5=0; G=0;
             if(X5){tm3=50; stan3=4;}
             else if(!tm3){tm3=30; stan3=3;}
             break;
    case 3: Z3=0; Z4=1; Z5=0; G=0;
             if(X5){tm3=50; stan3=4;}
             else if(!tm3){tm3=20; stan3=2;}
             break;
    case 4: Z3=0; Z4=0; Z5=0; G=1;
             if(T||!tm3){tm3=50; stan3=5;}
             break;
}
```

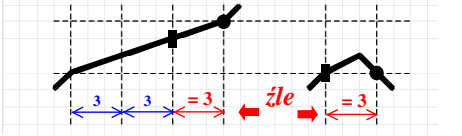
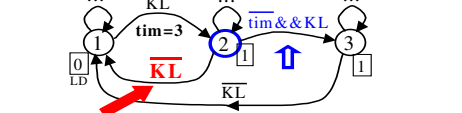
```

case 5: Z3=0; Z4=0; Z5=1; G=0;
      if(!tm3&&X7) stan3=6;
      else if(!tm3&&!X7) stan3=1;
      break;
}
if (tm3) --tm3;

L1=Z1; L2=Z2; L3=Z3; L4=Z4; L5=Z5; L6=G; // zaświeć diody

```

Najczęściej popełniane błędy to:

<p>Rysowanie na przebiegach czasowych, że się nalało lub wylało równy w 3 sekundy (przy cyklu 3+3). Ostatni cykl zawsze musi być "mniej niż" (nie może być "równy"). Obowiązuje to dla stanu opisanego jako "maksymalnie" jak też i "dokładnie" (nie ma znaczenia - zawsze "mniej niż").</p>	
<p>Napisanie w warunkach wychodzących ze stanu na jednej strzałce iloczynu (&&) z timerem (np. !tim&&KL), a na drugiej strzałce bez iloczynu z timerem (np. samo !KL). To jest bardzo poważny błąd w rozwiązaniu.</p>	
<p>Używanie tego samego timera (np. <i>tim</i>) w dwóch różnych grafach (instrukcjach switch), bo ustawienia zmiennej <i>tim</i> w pierwszym grafie będą zakłócać pracę drugiego grafu. Zawsze, każdy graf MUSI mieć osobne zmienne, np. w pierwszym <i>stan1</i>, <i>tim1</i> a w drugim grafie <i>stan2</i>, <i>tim2</i>.</p>	<pre> switch(stan1) { case 3: Z1=1; if (!tim) stan1=2; break; ... } switch(stan2) { case 5: Z2=1; if (!tim) stan2=1; break; ... } </pre>
<p>W sytuacji, gdy jednocześnie pracują oba timery, np. jak w zbiorniku 2, główny czas (tu 8s) może się skończyć tylko podczas odmierzenia pomocniczego czasu cyklu (tu 3s cyklu nalewania lub 2s pauzy) - oba te czasy nie mogą się skończyć jednocześnie (idealnie w tej samej chwili). W ostatnim cyklu nie mogą być oba jednocześnie "równy".</p>	