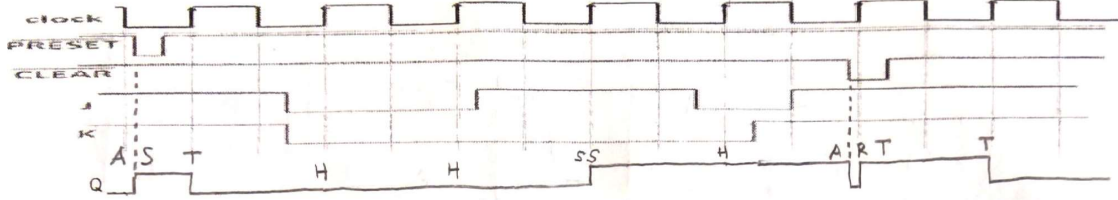
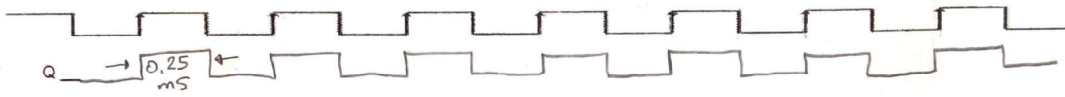


1	2	3	Toplam
1	3,4	3,4	PÖÇ

1. a. Draw output waveforms of a rising edge triggered J-K flip-flop for the input waveforms given below, indicating output transitions with **AS** Asynchronous Set, **AR** Asynchronous Reset, **SS** Synchronous Set, **SR** Synchronous Reset, **T** Toggle and **H** Hold. *Yükselen kenar tetiklemeli bir J-K flip flop'un çıkış dalga şeklini aşağıda verilen girişler için, çıkış değişimlerini, AS Asenkron Kurma, AR Asenkron Sıfırlama, SS Senkron Kurma, SR Senkron Sıfırlama, T Devrilme veya H tutma ile göstererek çiziniz.*

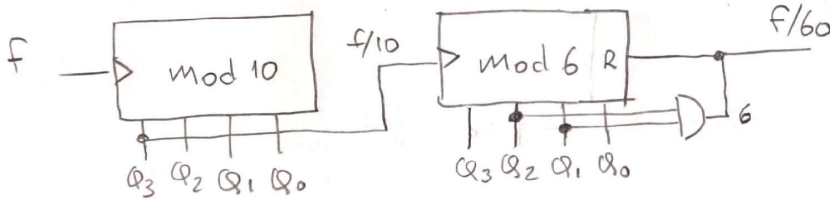


- b. Draw the input and output waveforms of a **retriggerable one-shot multivibrator** with $t_w = 0.25$ ms for $f = 2$ kHz trigger signal. *Yükselen kenar tetiklemeli bir J-K flip flop'un çıkış dalga şeklini aşağıda verilen girişler için, çıkış değişimlerini, AS Asenkron Kurma, AR Asenkron Sıfırlama, SS Senkron Kurma, SR Senkron Sıfırlama, T Devrilme veya H tutma ile göstererek çiziniz.*



$$f = 2 \text{ kHz} \quad T = 0.5 \text{ ms}$$

2. Use 4-bit asynchronous decade counter ICs to design a mod-60 counter to divide 60 Hz input frequency to generate 1 Hz clock pulse for a digital clock. *Bir sayısal saat için 60 Hz frekanslı girişi bölerek 1 Hz frekanslı saat işareti üreten mod-60 sayıcıyı 4-bit asenkron onlu sayıcı tümdevreleri kullanarak tasarlayınız.*



3. A Central Processing Unit (CPU) runs an algorithm in an infinite loop. The algorithm needs to execute the 0th, 1st, 2nd, 3rd and 4th microinstruction codes stored in the 0th, 1st, 2nd, 3rd and 4th address content of the microprogram memory by addressing these locations sequentially. Then the algorithm should skip the 5th, 6th, 7th and 8th microinstructions and continue executing the 9th, 10th and 11th microinstructions. After the 11th microinstruction is executed, the algorithm should jump back to the beginning of the loop and start executing 0th microinstruction again. Design an asynchronous counter circuit to generate address for the microprogram memory using flip-flops with active high asynchronous preset and clear inputs. The counter should be able to count 0-1-2-3-4-9-10-11-0-1-2-3-4-9-10-11-0-1-... continuously. Outputs of the counter with decimal values $2^0, 2^1, \dots$ should be named as $Q_0, Q_1, \dots$ and used directly with these signal names where necessary instead of connecting to the corresponding flip-flop outputs to make the design look simpler. *Merkezi İşlem Birimi (CPU), sonsuz bir döngüde bir algoritma çalıştırmaktadır. Algoritmanın, mikroprogram belleğinin 0., 1., 2., 3. ve 4. adresli içeriğinde depolanan 0., 1., 2., 3. ve 4. mikro emir kodlarını, bu konumları sırayla adresleyerek çalıştırması gerekir. Ardından algoritma 5., 6., 7. ve 8. mikro emirleri atlamalı ve 9., 10. ve 11. mikro emirleri çalıştırmaya devam etmelidir. 11. mikro emiri yürüttükten sonra, algoritma sonsuz döngünün başlangıcına geri dönmeli ve tekrar 0. mikro emiri çalıştırmaya başlamalıdır. Aktif yüksek asenkron kurma ve sıfırlama girişli flip-flop'ları kullanarak mikro program belleğine adres üretmek için asenkron bir sayaç devresini tasarlayınız. Sayaç 0-1-2-3-4-9-10-11-0-1-2-3-4-9-10-11-0-1-... olarak sürekli saymalıdır. Sayacın $2^0, 2^1, \dots$ değerli çıkışları, $Q_0, Q_1, \dots$ olarak adlandırılmalı ve tasarımın daha basit görünmesi için ilgili flip-flop çıkışlarına bağlanması yerine gerektiği yerlerde doğrudan bu işaret isimleri ile kullanılmalıdır.*

- a. How many flip-flops are needed to realize the counter? (6 points) *Sayacı gerçekleştirmek için kaç flip-flop kullanılmalıdır? (6 puan)*

$$\log_2 12 > 3 \Rightarrow 4$$

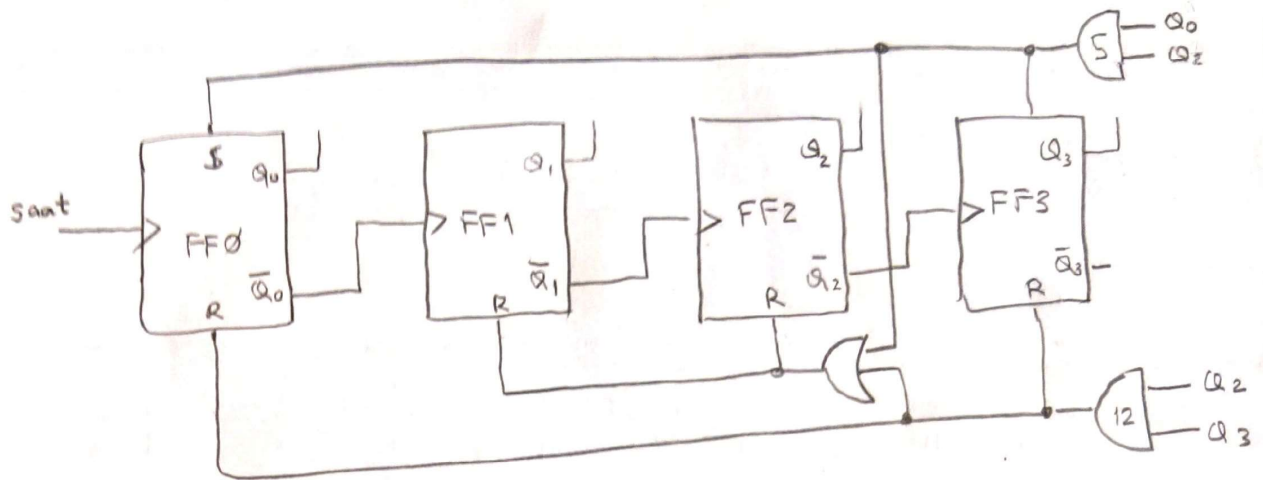
- b. How many decoders are needed to count the defined counting sequence and which values should these decoders decode? (6 points) *Tanımlanmış sayma sırasını saymak için kaç kod çözücüye ihtiyaç vardır ve bu kod çözücüler hangi değerlerin kodunu çözmelidir? (6 puan)*

4'den 9'a atlamak için 5'in kodunu çözen ve 11'den 0'a atlamak için 12'nin kodunu çözen

- c. How should these decoder outputs be combined to generate required asynchronous preset/reset signals to flip-flops? (8 points) *Flip-flop'lara gerekli asenkron kurma/sıfırlama işaretlerini üretmek için bu kod çözücü çıkışları nasıl birleştirilmelidir? (8 puan)*

5'i algılayan kod çözücü 9 için FF0 ve FF3'ü kurmalı FF1 ve FF2'yi sıfırlamalı.
12'yi " " " " 0 için tüm FF'leri sıfırlamalı.
FF'lerin asenkron kurma ve sıfırlama girişleri ayrı olduğu için ve 12'tüm FF'leri ve 5 FF1 ve FF2'yi sıfırlaması gerektiği için FF1 ve FF2'nin sıfırlama girişleri 5 ve 12 çıkışlarının birleştirildiği bir VEYA kapısı çıkışına bağlanmalıdır.

d. Draw the counter using flip-flops and gates. (16 points) /?/ Sayacı flip-flop ve kapılar kullanarak çiziniz. (16 puan)



e. Draw the output waveforms of the counter, showing decoded preset/reset signals and show that it is indeed counting as defined. (12 points) /?/ Kodu çözülmüş kurma/sıfırlama işaretlerini ve sayacın çıkışlarını gösteren dalga şekillerini çizin ve gerçekten tanımlandığı gibi saydığını gösterin. (12 puan)

